# HarvardX Data Science Capstone Project

London Crime Prediction

*Erdmuthe Ellmann*

*May 31st 2020*

## Contents

# 1 Introduction

This project is part of the ninth and final course in HarvardX's multi-part Data Science Professional Certificate series (PH125x) and intends to apply all the tools learned throughout the series. In this final project we were allowed to choose the prediction problem and dataset ourselves. As I'm a great fan of crime fiction and also find forensic science and analytics absolutely fascinating I decided to analyze the London Crime dataset which is publicly available on Kaggle (source: https://www.kaggle.com/jboysen/london-crime, accessed on 04/05/2020). To make the prediction problem a bit more challenging I've added another dataset, London Boroughs, with socio-demographic data for the boroughs of London (source: https://www.kaggle.com/marshald/london-boroughs, accessed on 04/10/2020). Unlike London Crime, this dataset is not an ML-friendly or cleaned dataset and needed some data wrangling before I could get started with the analysis and predictions.

The goal of this project is to generate predictions for the number of crimes of the different boroughs. The socio-demographic information of these boroughs, like population density, average age or employment rate, will be used to predict the number of crimes.

First of all, we will start out with the data preparation needed to create the cleaned and combined dataset. This will be followed by an inital data exploration to get an overview and first impression of the dataset. Next, we will dive into a detailed exploratory analysis of the four different groups of variables and figure out which of them are the most promising predictors that we should later include in our prediction model. During the development of the prediction, we will try different models and combinations of predictors in order to get to the best suited model. We will start with the simple Average as a base prediction, followed by Linear Regression. Then we will try two more advanced algorithms which are K-Nearest Neighbours and a Regression Tree using the `rpart` package. Forecast accuracy will be evaluated based on the Residual Mean Square Error (RMSE). Finally, a conclusion will briefly summarize the key findings and limitations of the project.

# 2 Data Preparation

As we have two datasets, one already ML-friendly and clean and one first needing some cleaning, severeal data preparation steps need to be performed before we can actually start the analysis.

## 2.1 Create the Combined Dataset

The London Crime dataset with it's original content contains seven variables (excluding the first column, which is just an index, see table below). Two of the variables are about the area the crime happened (`lsoa_code` and `borough`), two describing the crime (`major_category` and `minor_category`), two about the `month` and `year` the reported crime happened and lastly the number of crimes (`value`).

| X | lsoa_code | borough | major_category | minor_category | value | year | month |
|---|---|---|---|---|---|---|---|
| 1 | E01001116 | Croydon | Burglary | Burglary in Other Buildings | 0 | 2016 | 11 |
| 2 | E01001646 | Greenwich | Violence Against the Person | Other violence | 0 | 2016 | 11 |
| 3 | E01003774 | Redbridge | Burglary | Burglary in Other Buildings | 0 | 2016 | 3 |
| 4 | E01004177 | Sutton | Theft and Handling | Theft/Taking of Pedal Cycle | 1 | 2016 | 8 |
| 5 | E01002398 | Hillingdon | Theft and Handling | Theft/Taking Of Motor Vehicle | 0 | 2016 | 2 |
| 6 | E01002945 | Kingston upon Thames | Theft and Handling | Theft From Shops | 0 | 2016 | 11 |

This data is available for 2008 to 2016. I thought it would be very interesting to find out more about the single boroughs and the population that lives there in order to gain more insights into the crime occurences and predict them. Therefore I searched for socio-demographic data of the boroughs of London which I could add to the dataset. The London Boroughs dataset I found only has data from 2016, apart from the employment data, which is from 2015. Surely it would have been even better if the socio-demographic data would have been available for a longer time period to analyze several years, however, I know from experience that a perfect database is rarely available in real-life problems. Thus, I decided to go with this anyways and drop the years 2008 to 2015 from the London Crime dataset in favor of having a larger pool of variables by adding the London Boroughs dataset. This means we will focus on the year 2016 only.

The London Crime dataset is already clean, so we only have to do a little bit of data preparation. I've added a column combining year and month, and then aggregated the dataset without the first two columns, as the first column is just an index and the second column is the `lsoa_code`, which we don't need. LSOA is the abbreviation for "Lower Super Output Area" and basically represents a more granular division of the Metropolitan Area of London than the boroughs. Due to the fact that we are going to predict crimes per borough, we don't need this variable and can downsize the dataset with this aggregation. The resulting dataset is shown in the table below.

```
# add a new column to londoncrime which combines year and month
londoncrime_temp$date <- with(londoncrime_temp, sprintf("%d-%02d", year, month))

# aggregate the dataset without the first two columns
# (one is just an index and lsoa_code is not needed and just makes the dataset much bigger)
londoncrime_temp <- londoncrime_temp %>%
  group_by(borough, major_category, minor_category, year, month, date) %>%
  summarize(value = sum(value)) %>%
  select(borough, major_category, minor_category, value, year, month, date)

# change class into tibble
londoncrime_temp <- as_tibble(londoncrime_temp)

# check the resulting dataset
```

```
head(londoncrime_temp) %>%
  kable(booktabs = TRUE) %>%
  kable_styling(latex_options = c("striped"),
                position = "center",
                font_size = 9,
                full_width = FALSE)
```

| borough | major_category | minor_category | value | year | month | date |
|---|---|---|---|---|---|---|
| Barking and Dagenham | Burglary | Burglary in a Dwelling | 103 | 2016 | 1 | 2016-01 |
| Barking and Dagenham | Burglary | Burglary in a Dwelling | 86 | 2016 | 2 | 2016-02 |
| Barking and Dagenham | Burglary | Burglary in a Dwelling | 72 | 2016 | 3 | 2016-03 |
| Barking and Dagenham | Burglary | Burglary in a Dwelling | 47 | 2016 | 4 | 2016-04 |
| Barking and Dagenham | Burglary | Burglary in a Dwelling | 77 | 2016 | 5 | 2016-05 |
| Barking and Dagenham | Burglary | Burglary in a Dwelling | 63 | 2016 | 6 | 2016-06 |

The London Boroughs dataset has 40 rows and 85 columns, which include a large variety of different variables, ranging from population size to age, employment data and even waste recycling rates and turnouts at local elections. The table shows an exerpt of some selected variables.

```
# check dimensions of london boroughs dataset
dim(londonboroughs_temp)
```

```
## [1] 40 85
```

| New.code | Area.name | Inner..Outer.London | GLA.Population.Estimate.2016 | GLA.Household.Estimate.2016 |
|---|---|---|---|---|
| E09000001 | City of London | Inner London | 8,548 | 5,179 |
| E09000002 | Barking and Dagenham | Outer London | 205,773 | 76,841 |
| E09000003 | Barnet | Outer London | 385,108 | 149,147 |
| E09000004 | Bexley | Outer London | 243,303 | 97,233 |
| E09000005 | Brent | Outer London | 328,568 | 119,166 |

| Average.Age..2016 | Employment.rate.....2015. | Household.Waste.Recycling.Rate..2014.15 | Turnout.at.2014.local.elections |
|---|---|---|---|
| NA | NA | | |
| 42.9 | 64.6 | 34.4 | . |
| 32.9 | 65.8 | 23.4 | 36.5 |
| 37.2 | 68.5 | 38.0 | 40.5 |
| 38.9 | 75.1 | 54.0 | 39.6 |
| 35.5 | 69.5 | 35.2 | 36.3 |

It is clear that this dataset needs a bit more preparation, as there are empty rows, NAs, whitespaces infront of numbers, unfavorable headings and some totals for England, United Kingdom, Inner and Outer London added to the borough column. However, once this is cleaned, we will have some very interesting socio-demographic information for every borough in the London Crime dataset.

First of all, we are going to pick out the most interesting variables, as analyzing all 85 in detail would go beyond the scope of this project. We can also rename them, remove the first row, which is blank, and remove the last six rows containing additional data for total UK, England et cetera. Furthermore we replace the dots with the value "NA", remove the whitespace infront of the numbers and change the class to numeric. The resulting dataset is shown in the table below.

```r
# extract the relevant columns from the dataset and rename them
londonboroughs <- londonboroughs_temp %>%
  select(Area.name, GLA.Population.Estimate.2016,
         GLA.Household.Estimate.2016, Inland.Area..Hectares.,
         Population.density..per.hectare..2016, Average.Age..2016,
         Proportion.of.population.aged.0.15..2016,
         Proportion.of.population.of.working.age..2016,
         Proportion.of.population.aged.65.and.over..2016,
         Employment.rate......2015., Male.employment.rate..2015.,
         Female.employment.rate..2015.,
         Unemployment.rate..2015.) %>%
  setnames(c("Area.name", "GLA.Population.Estimate.2016",
             "GLA.Household.Estimate.2016", "Inland.Area..Hectares.",
             "Population.density..per.hectare..2016", "Average.Age..2016",
             "Proportion.of.population.aged.0.15..2016",
             "Proportion.of.population.of.working.age..2016",
             "Proportion.of.population.aged.65.and.over..2016",
             "Employment.rate......2015.", "Male.employment.rate..2015.",
             "Female.employment.rate..2015.",
             "Unemployment.rate..2015."),
           c("borough", "population", "household", "hectares", "population_density",
             "avg_age", "population_under_16", "population_working_age", "population_over_64",
             "employment_rate_2015", "employment_rate_male_2015", "employment_rate_female_2015",
             "unemployment_rate_2015"))

# remove the first row (this row is blank)
londonboroughs <- londonboroughs[-1,]

# remove the last 6 rows (these rows contain additional data for total UK, England etc., which we
# don't need)
londonboroughs <- londonboroughs[-c(34:39),]

# replace the "." with "NA"
londonboroughs[londonboroughs == "."] <- NA

# remove whitespace infront of the numbers in the columns population and household
londonboroughs <- londonboroughs %>%
  mutate_at(2:3, str_trim)

# set format to numeric for the columns which are saved as characters at the moment (parse_number
# removes the commas and converts to numeric)
londonboroughs <- londonboroughs %>%
  mutate_at(2:5, parse_number)

londonboroughs <- londonboroughs %>%
  mutate_at(11:13, parse_number)

# check the resulting dataset
str(londonboroughs)
```

```
## 'data.frame':    33 obs. of  13 variables:
##  $ borough                   : chr  "City of London" "Barking and Dagenham" "Barnet" "Bexley" ...
##  $ population                : num  8548 205773 385108 243303 328568 ...
```

```
##  $ household                   : num  5179 76841 149147 97233 119166 ...
##  $ hectares                    : num  290 3611 8675 6058 4323 ...
##  $ population_density          : num  28.9 57.3 44.5 39.9 76.1 ...
##  $ avg_age                     : num  42.9 32.9 37.2 38.9 35.5 40.1 36.2 36.9 36.1 36.2 ...
##  $ population_under_16         : num  27.2 21 21 20.8 20.1 15.8 8.8 22.3 21.2 22.7 ...
##  $ population_working_age      : num  90.6 86.1 83.3 89 82.5 88.4 82.8 87 88.3 87.2 ...
##  $ population_over_64          : num  9.4 13.9 16.7 11 17.5 11.6 17.2 13 11.7 12.8 ...
##  $ employment_rate_2015        : num  64.6 65.8 68.5 75.1 69.5 75.3 69.2 75.4 72.7 73 ...
##  $ employment_rate_male_2015   : num  NA 75.6 74.5 82.1 76 80.4 72.2 81.8 81.2 80.4 ...
##  $ employment_rate_female_2015: num  NA 56.5 62.9 68.5 62.6 70.4 66.1 69.5 63.8 66 ...
##  $ unemployment_rate_2015      : num  NA 11 8.5 7.6 7.5 5.3 4 4.1 5.8 3.8 ...
```

| borough | population | household | hectares | population_density | avg_age | population_under_16 | population_working_age |
|---|---|---|---|---|---|---|---|
| City of London | 8548 | 5179 | 290.4 | 28.9 | 42.9 | 27.2 | 90.6 |
| Barking and Dagenham | 205773 | 76841 | 3610.8 | 57.3 | 32.9 | 21.0 | 86.1 |
| Barnet | 385108 | 149147 | 8674.8 | 44.5 | 37.2 | 21.0 | 83.3 |
| Bexley | 243303 | 97233 | 6058.1 | 39.9 | 38.9 | 20.8 | 89.0 |
| Brent | 328568 | 119166 | 4323.3 | 76.1 | 35.5 | 20.1 | 82.5 |
| Bromley | 326560 | 139654 | 15013.5 | 21.7 | 40.1 | 15.8 | 88.4 |

| population_over_64 | employment_rate_2015 | employment_rate_male_2015 | employment_rate_female_2015 | unemployment_rate_2015 |
|---|---|---|---|---|
| 9.4 | 64.6 | NA | NA | NA |
| 13.9 | 65.8 | 75.6 | 56.5 | 11.0 |
| 16.7 | 68.5 | 74.5 | 62.9 | 8.5 |
| 11.0 | 75.1 | 82.1 | 68.5 | 7.6 |
| 17.5 | 69.5 | 76.0 | 62.6 | 7.5 |
| 11.6 | 75.3 | 80.4 | 70.4 | 5.3 |

Now that we have both datasets cleaned and in the same format, we can just combine them.

```
# add the information of the londonboroughs dataset to the londoncrime dataset
londoncrime <- left_join(londoncrime_temp, londonboroughs, by = "borough")

# check the resulting dataset
str(londoncrime)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    12552 obs. of  19 variables:
##  $ borough                   : chr  "Barking and Dagenham" "Barking and Dagenham" "Barking and Dage
##  $ major_category            : chr  "Burglary" "Burglary" "Burglary" "Burglary" ...
##  $ minor_category            : chr  "Burglary in a Dwelling" "Burglary in a Dwelling" "Burglary in a
##  $ value                     : int  103 86 72 47 77 63 54 54 48 60 ...
##  $ year                      : int  2016 2016 2016 2016 2016 2016 2016 2016 2016 2016 ...
##  $ month                     : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ date                      : chr  "2016-01" "2016-02" "2016-03" "2016-04" ...
##  $ population                : num  205773 205773 205773 205773 205773 ...
##  $ household                 : num  76841 76841 76841 76841 76841 ...
##  $ hectares                  : num  3611 3611 3611 3611 3611 ...
##  $ population_density        : num  57.3 57.3 57.3 57.3 57.3 57.3 57.3 57.3 57.3 57.3 ...
##  $ avg_age                   : num  32.9 32.9 32.9 32.9 32.9 32.9 32.9 32.9 32.9 32.9 ...
##  $ population_under_16       : num  21 21 21 21 21 21 21 21 21 21 ...
##  $ population_working_age    : num  86.1 86.1 86.1 86.1 86.1 86.1 86.1 86.1 86.1 86.1 ...
##  $ population_over_64        : num  13.9 13.9 13.9 13.9 13.9 13.9 13.9 13.9 13.9 13.9 ...
```

```
##  $ employment_rate_2015       : num  65.8 65.8 65.8 65.8 65.8 65.8 65.8 65.8 65.8 65.8 ...
##  $ employment_rate_male_2015  : num  75.6 75.6 75.6 75.6 75.6 75.6 75.6 75.6 75.6 75.6 ...
##  $ employment_rate_female_2015: num  56.5 56.5 56.5 56.5 56.5 56.5 56.5 56.5 56.5 56.5 ...
##  $ unemployment_rate_2015     : num  11 11 11 11 11 11 11 11 11 11 ...
```

```
# remove the temporary datasets
rm(londonboroughs_temp, londoncrime_temp)
```

## 2.2 Missing Value for City of London

Unfortunately, there is one missing value in our dataset. There is no `employment_rate_male_2015` given for the borough "City of London", but we will later need this variable for our prediction model. Due to the fact that we only have a total of 33 boroughs, we don't really want to just remove this borough from the dataset. So how should we deal with the missing value?

| borough | employment_rate_2015 | employment_rate_male_2015 | delta | delta_prcnt |
|---|---|---|---|---|
| Barking and Dagenham | 65.8 | 75.6 | 9.8 | 14.9 |
| Barnet | 68.5 | 74.5 | 6.0 | 8.8 |
| Bexley | 75.1 | 82.1 | 7.0 | 9.3 |
| Brent | 69.5 | 76.0 | 6.5 | 9.4 |
| Bromley | 75.3 | 80.4 | 5.1 | 6.8 |
| Camden | 69.2 | 72.2 | 3.0 | 4.3 |
| City of London | 64.6 | NA | NA | NA |
| Croydon | 75.4 | 81.8 | 6.4 | 8.5 |
| Ealing | 72.7 | 81.2 | 8.5 | 11.7 |

As we said, we only have a small number of boroughs, so just excluding this borough from the analysis seems like an unfavourable solution. Unfortunately it is completely unclear why this value is missing from the dataset. Is there a reason the employment rate can't be reported for the City of London? Is it not missing at random, but depends on the hypothetical value (e.g. similar to a logic like "missing value for salary -> people with high salaries generally do not want to reveal their incomes in surveys")? Does the City of London have similar characteristics to certain other boroughs which we could use for forming an assumption? Would we introduce bias by excluding this borough or forming an unfortunate assumption for the employment rate? I've researched how to deal with this problem on the internet and a very nice overview of different approaches can be found on towardsdatascience.com (https://towardsdatascience.com/how-to-handle-missing-data-8646b18db0d4, accessed on 04/19/2020).

For the problem at hand I've decided on the following:

- We will form an assumption for the employment rate male for borough City of London rather than excluding it.
- The table above shows that for all other boroughs the employment rate male is higher than the employment rate overall (male & female). We will use this insight to derive an assumption for the employment rate male based on the overall employment rate for City of London.
- The prediction we will use is `employment_rate_2015` for City of London + average `delta_prcnt` that the other boroughs show (`delta_prcnt` = difference between male and overall employment rate in percent)

| borough | employment_rate_2015 | employment_rate_male_2015 |
|---|---|---|
| Barking and Dagenham | 65.8 | 75.6 |
| Barnet | 68.5 | 74.5 |
| Bexley | 75.1 | 82.1 |
| Brent | 69.5 | 76.0 |
| Bromley | 75.3 | 80.4 |
| Camden | 69.2 | 72.2 |
| City of London | 64.6 | 70.3 |
| Croydon | 75.4 | 81.8 |
| Ealing | 72.7 | 81.2 |

The resulting prediction for the `employment_rate_male_2015` for City of London is 70.3.

## 2.3 Split the Dataset into Train, Test and Validation Sets

Similar to the validation in the MovieLens Project, we will first seperate a validation dataset and then split the remaining dataset into a train and test set. This means we can perform full cross-validation. The validation dataset will only be used at the very end of this project to evaluate the performance of the final algorithm. During model building we will use the train dataset to train each algorithm and use the test dataset to tune parameters and evaluate which algorithm seems most promising.

We have two conflicting goals here: on the one hand, we would like to have the largest possible database to train our model with, because this should improve our model performance. On the other hand, our test or validation datasets should not be too small, otherwise we will get very imprecise performance results, because we are just evaluating against a handful, maybe very uncommon data records. Common approaches for splits are 80:20, 90:10 or even 50:50. Considering that we want to split into three datasets this would translate into 60/20/20, 80/10/10 and 33/33/33 as a proportion for the train/test/validation set. In the problem at hand we have a total of 33 boroughs, so 10% of the boroughs would be just 3 boroughs (rounded, as we have to look at each borough as a whole and can't split it). This seems to be a very small number for evaluation. On the other hand, 33% of the boroughs would be 11 boroughs, which would be a very small number for training. Thus, a split of 60/20/20 seems to be the most favourable option in this case. This means we will split our dataset into 6 randomly selected boroughs for validation and another 6 randomly selected for testing, which leaves 21 boroughs for trainig. Now we are all set for starting with the exploratory data analysis (EDA) on the training dataset in the next chapter.

```
# create auxiliary table for boroughs
boroughs <- londoncrime %>%
  group_by(borough) %>%
  summarize(crimes = sum(value))

# randomly select 6 of the 33 boroughs
set.seed(9)
index_validation <- boroughs[sample(nrow(boroughs), 6), ] %>%
  .$borough

# seperate the data of the randonly selected boroughs to get the validation dataset
validation <- londoncrime %>%
  filter(borough %in% index_validation)

# remove the boroughs in the validation set from the list of boroughs
boroughs <- boroughs %>%
```

```
  filter(!borough %in% index_validation)

# then do the same to split the remaining dataset into a train and test set we can use to build
# our model
index_test <- boroughs[sample(nrow(boroughs), 6), ] %>%
  .$borough

test <- londoncrime %>%
  filter(borough %in% index_test)

# create the train set with the rest of the data
train <- londoncrime %>%
  filter(!borough %in% c(index_validation, index_test))

# check that separation went fine
nrow(test) + nrow(train) + nrow(validation) == nrow(londoncrime)
```

```
## [1] TRUE
```

```
# double check that split into validation, train and test set is ok as far as % of the original
# dataset is concerned -> both make up about 20% of the original london crime dataset
nrow(validation) / (nrow(londoncrime)) * 100
```

```
## [1] 17.39962
```

```
nrow(test) / (nrow(londoncrime)) * 100
```

```
## [1] 18.35564
```

```
nrow(test) / (nrow(train) + nrow(test)) * 100
```

```
## [1] 22.22222
```

# 3 Exploratory Data Analysis

To fully understand the dataset we will now explore all variables and also try to figure out correlations between them. The insights gained in this chapter will help us select the right prediction methods for our problem as well as the most important predictors to include. The goal is to predict number of crimes per month and borough, so we need to look for factors that influence the number of crimes. Analytics has been used in criminology for a long time. The problem at hand surely is a rather small prediction problem, however, it could still serve a practical purpose. Like any other organisation, the police also has to do mid- and long-term resource planning, and predicting the development of the number of crimes over the next years could help to estimate how many police resources will be needed and where to best put these limited resources. Surely, the police has experience with crime occurences and the severe crime hot-spots are already known, but a large metropolitan area like London is constantly evolving and you would want to discover new trends and changing characteristics as early as possible to be able to react appropriately.

## 3.1 Get a First Impression of the Dataset

The train dataset contains a bit more than 8.000 data records and 19 different variables. We can see the name of the borough, the major and minor category of the crime, the number of crimes (`value`) as well as the year and month the crime was reported and the combined variable of year and month which we've added in the data preparation. The next four variables are GLA 2016-based population estimates and show the population size living in each borough, the number of households, the hectares of the borough and the population density (per hectare). Furthermore, we have another four variables about the age of the borough population, which include average age, proportion of the population with an age under 16 (age 0-15), population proportion in working age (age 16-64) and population proportion over 64 (age 65 and over). Finally, there are another four variables about the labour market, the overall employment rate, the male employment rate, the female employment rate and the unemployment rate. These four variables are from 2015, whereas all other variables are from 2016.

| borough | major_category | minor_category | value | year | month | date | population | household |
|---|---|---|---|---|---|---|---|---|
| Bexley | Burglary | Burglary in a Dwelling | 86 | 2016 | 1 | 2016-01 | 243303 | 97233 |
| Bexley | Burglary | Burglary in a Dwelling | 73 | 2016 | 2 | 2016-02 | 243303 | 97233 |
| Bexley | Burglary | Burglary in a Dwelling | 57 | 2016 | 3 | 2016-03 | 243303 | 97233 |
| Bexley | Burglary | Burglary in a Dwelling | 31 | 2016 | 4 | 2016-04 | 243303 | 97233 |
| Bexley | Burglary | Burglary in a Dwelling | 30 | 2016 | 5 | 2016-05 | 243303 | 97233 |
| Bexley | Burglary | Burglary in a Dwelling | 38 | 2016 | 6 | 2016-06 | 243303 | 97233 |

| hectares | population_density | avg_age | population_under_16 | population_working_age | population_over_64 |
|---|---|---|---|---|---|
| 6058.1 | 39.9 | 38.9 | 20.8 | 89 | 11 |
| 6058.1 | 39.9 | 38.9 | 20.8 | 89 | 11 |
| 6058.1 | 39.9 | 38.9 | 20.8 | 89 | 11 |
| 6058.1 | 39.9 | 38.9 | 20.8 | 89 | 11 |
| 6058.1 | 39.9 | 38.9 | 20.8 | 89 | 11 |
| 6058.1 | 39.9 | 38.9 | 20.8 | 89 | 11 |

| employment_rate_2015 | employment_rate_male_2015 | employment_rate_female_2015 | unemployment_rate_2015 |
|---|---|---|---|
| 75.1 | 82.1 | 68.5 | 7.6 |
| 75.1 | 82.1 | 68.5 | 7.6 |
| 75.1 | 82.1 | 68.5 | 7.6 |
| 75.1 | 82.1 | 68.5 | 7.6 |
| 75.1 | 82.1 | 68.5 | 7.6 |
| 75.1 | 82.1 | 68.5 | 7.6 |

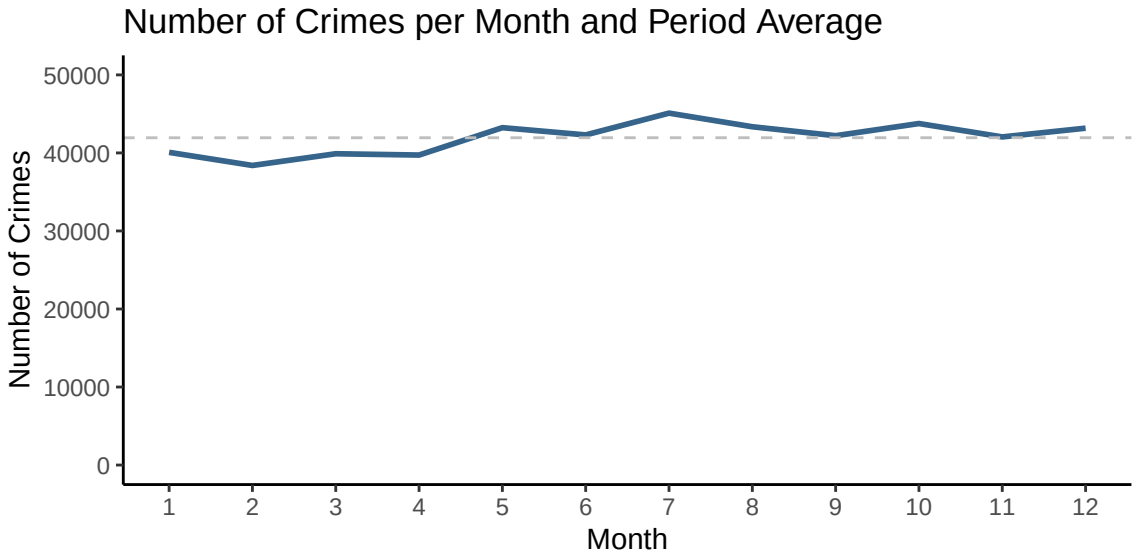
The table below shows the number of distinct major and minor categories as well as boroughs. Underneath are tables with all unique values for these three variables.

| Number of Major Categories | Number of Minor Categories | Number of Boroughs |
|---|---|---|
| 9 | 32 | 21 |

| Major Category | Minor Category |
|---|---|
| Burglary | Burglary in a Dwelling |
| Burglary | Burglary in Other Buildings |
| Criminal Damage | Criminal Damage To Dwelling |
| Criminal Damage | Criminal Damage To Motor Vehicle |
| Criminal Damage | Criminal Damage To Other Building |
| Criminal Damage | Other Criminal Damage |
| Drugs | Drug Trafficking |
| Drugs | Other Drugs |
| Drugs | Possession Of Drugs |
| Fraud or Forgery | Counted per Victim |
| Fraud or Forgery | Other Fraud & Forgery |
| Other Notifiable Offences | Going Equipped |
| Other Notifiable Offences | Other Notifiable |
| Robbery | Business Property |
| Robbery | Personal Property |
| Sexual Offences | Other Sexual |
| Sexual Offences | Rape |
| Theft and Handling | Handling Stolen Goods |
| Theft and Handling | Motor Vehicle Interference & Tampering |
| Theft and Handling | Other Theft |
| Theft and Handling | Other Theft Person |
| Theft and Handling | Theft From Motor Vehicle |
| Theft and Handling | Theft From Shops |
| Theft and Handling | Theft/Taking Of Motor Vehicle |
| Theft and Handling | Theft/Taking of Pedal Cycle |
| Violence Against the Person | Assault with Injury |
| Violence Against the Person | Common Assault |
| Violence Against the Person | Harassment |
| Violence Against the Person | Murder |
| Violence Against the Person | Offensive Weapon |
| Violence Against the Person | Other violence |
| Violence Against the Person | Wounding/GBH |

| Borough |
|---|
| Bexley |
| Camden |
| Ealing |
| Enfield |
| Greenwich |
| Hackney |
| Haringey |
| Hillingdon |
| Hounslow |
| Islington |
| Kensington and Chelsea |
| Kingston upon Thames |
| Lewisham |
| Merton |
| Newham |
| Redbridge |
| Richmond upon Thames |
| Southwark |
| Tower Hamlets |
| Wandsworth |
| Westminster |

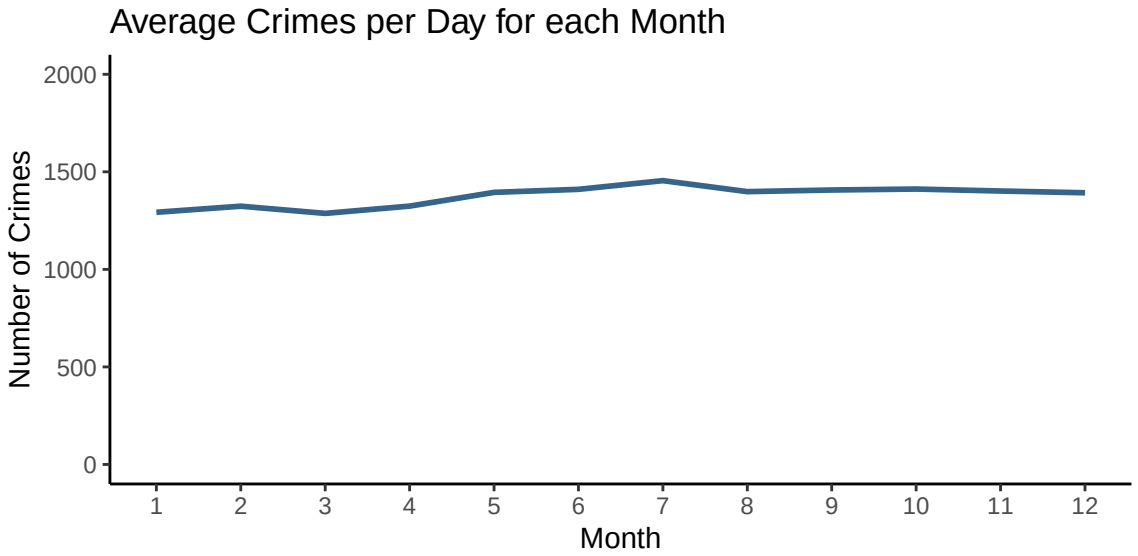
Over all boroughs, there are 41940 crimes reported per month on average. The chart shown below illustrates the number of crimes per month compared to this average. The number of crimes per month vary a bit with higher Jan, Mar, May, Jul, Oct and Dec versus lower Feb, Apr, Jun, Aug, Sep and Nov. Overall there seems to be an increasing trend as well, as Jan to Apr are below average, but all months after May are above average.

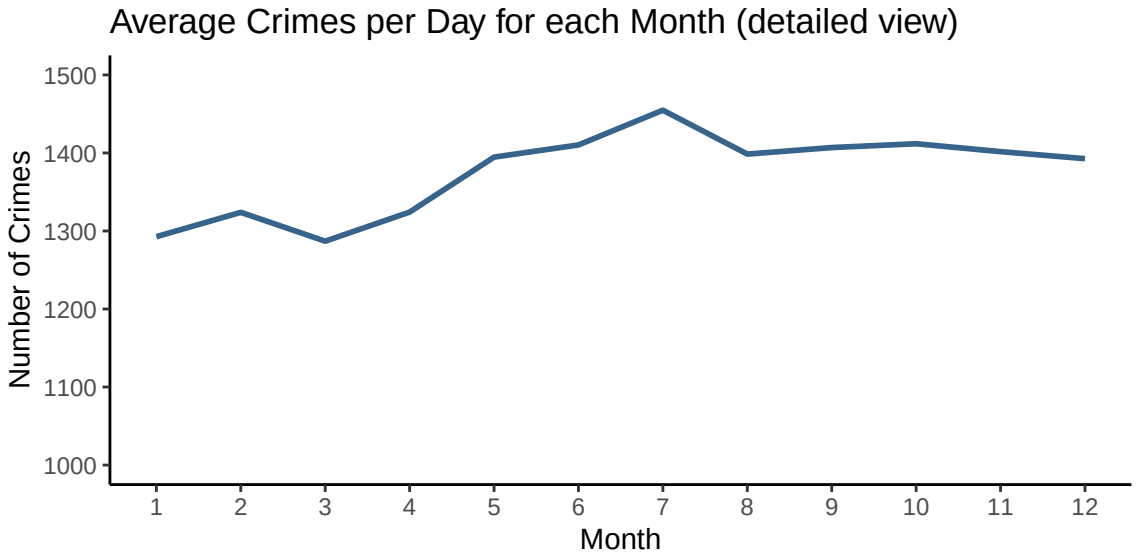



Could the monthly up and down pattern have to do with different number of days per month?

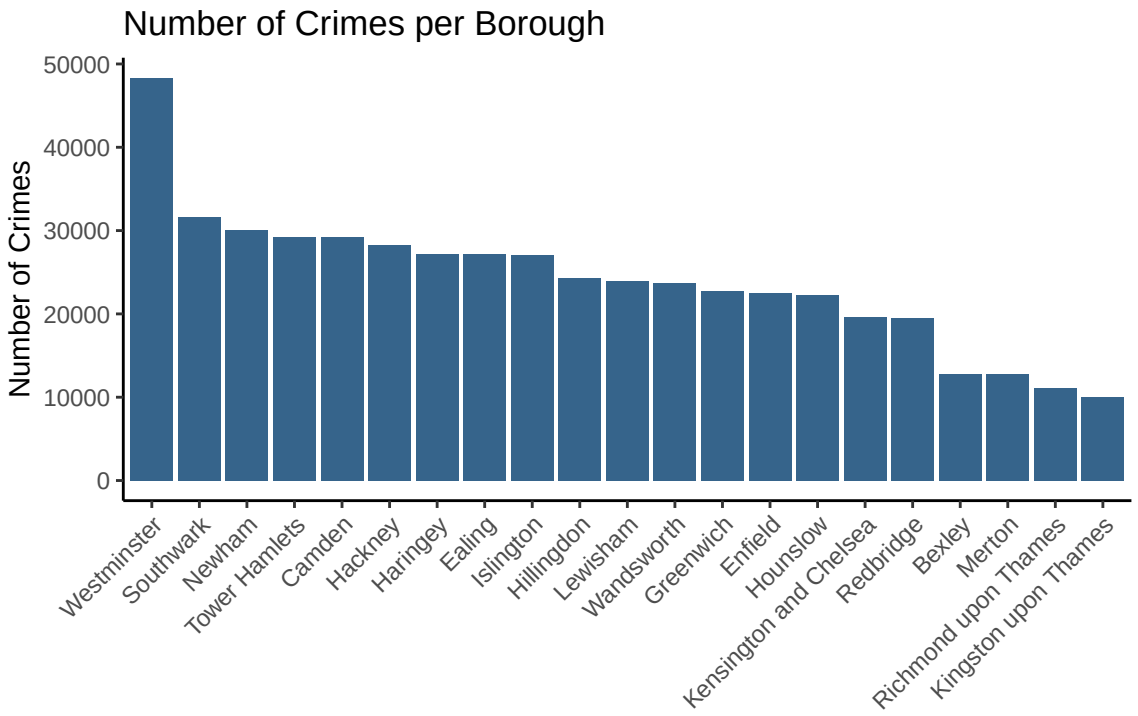
Looking at the charts with average crimes per day, one with the y-axis starting at 0, one displaying more detail with the y-axis starting at 1.000, we can see that some of the variation seems to have levelled out if one considers the different number of days. Now the pattern more looks like it could be driven by school holidays (e.g. Easter in Mar/Apr, summer holidays in Aug) and seasonality, maybe due to weather conditions.

However, this is not the focus of the project and we don't have the necessary data to further investigate this, so we'll leave it at that.

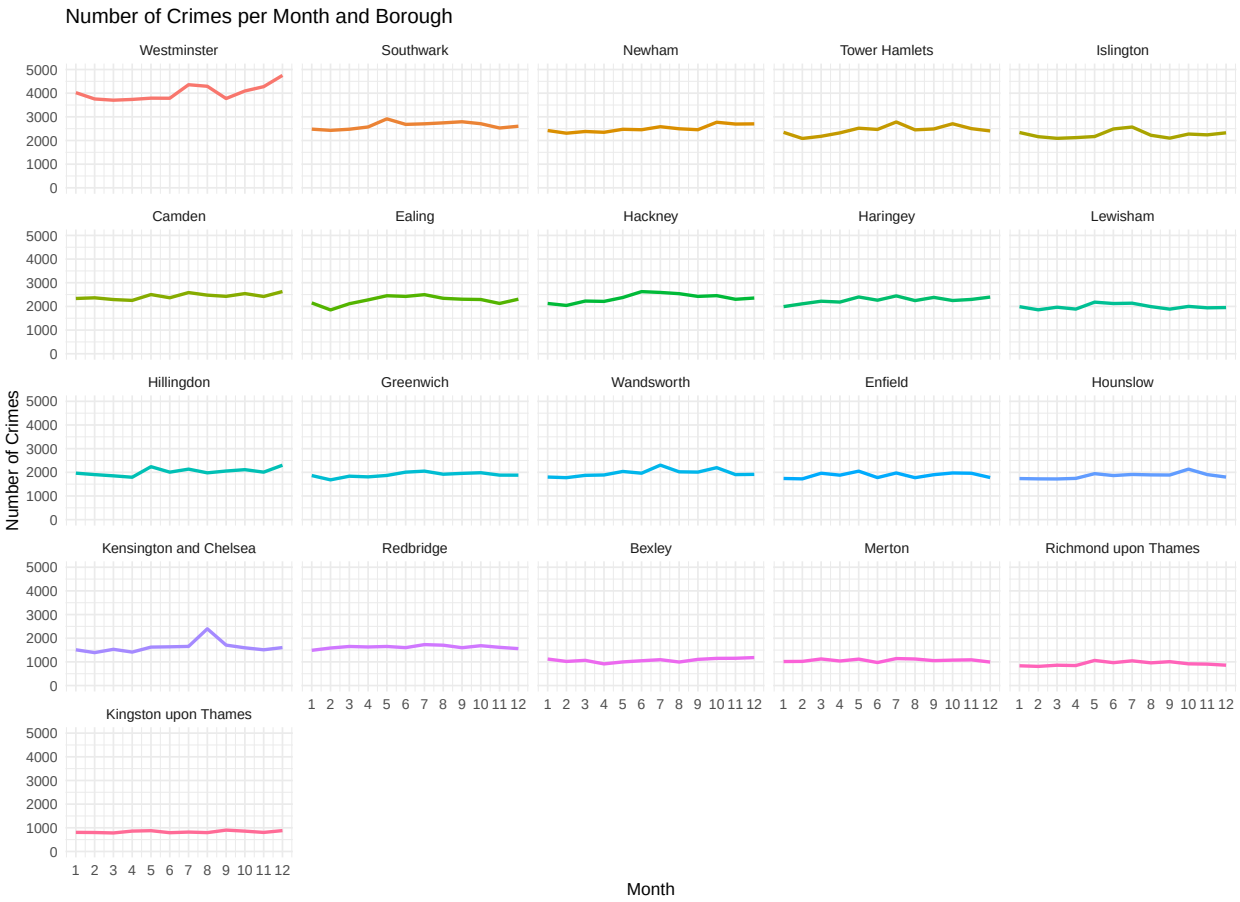



Westminster is the borough with the highest number of crimes, followed by Southwark and Newham. Kingston upon Thames has the lowest number of crimes in our train dataset.




The number of crimes in each borough seems to be relatively stable over those 12 months. This suggests that linear regression should be a good prediction method to start with. Only single spikes, like in Westminster in Jul, Aug and Dec are striking. However, boroughs seem to be quite different, some range around 1.000 crimes per month, some show much higher numbers, e.g. Westminster which ranges around 4.000 crimes per month.




## 3.2 Explore the Dataset in Detail

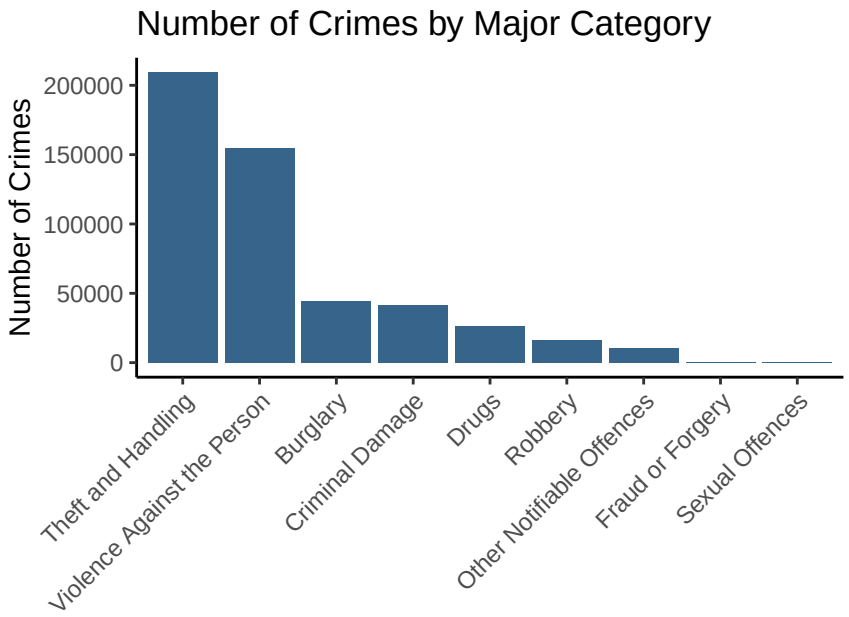
Now that we've gotten a first impression of the dataset, we will throughly analyze the different sets of variables. The goal is to 1) get a good understanding of the dataset and 2) find insights like a correlation between a variable and the number of crimes, so we know which variables we should later include in our prediction model.

The variables can be grouped in the following manner:

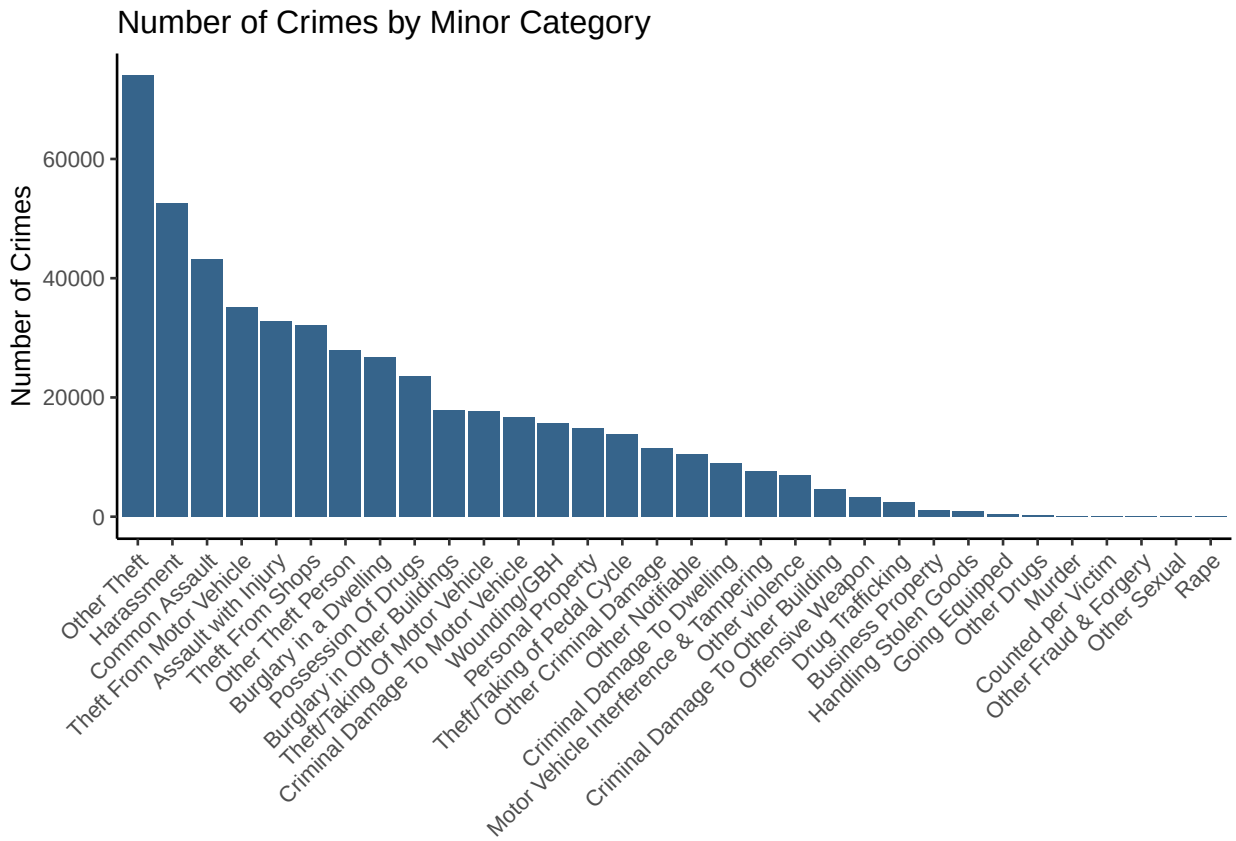
- Type of Crime: Major and Minor Category
- Population Size: Population, Household, Hectares and Population Density
- Population Age: Average Age, Population under 16, Population in Working Age and Population over 64
- Employment: Employment Rate, Employment Rate Male, Employment Rate Female and Unemployment Rate

### 3.2.1 Type of Crime

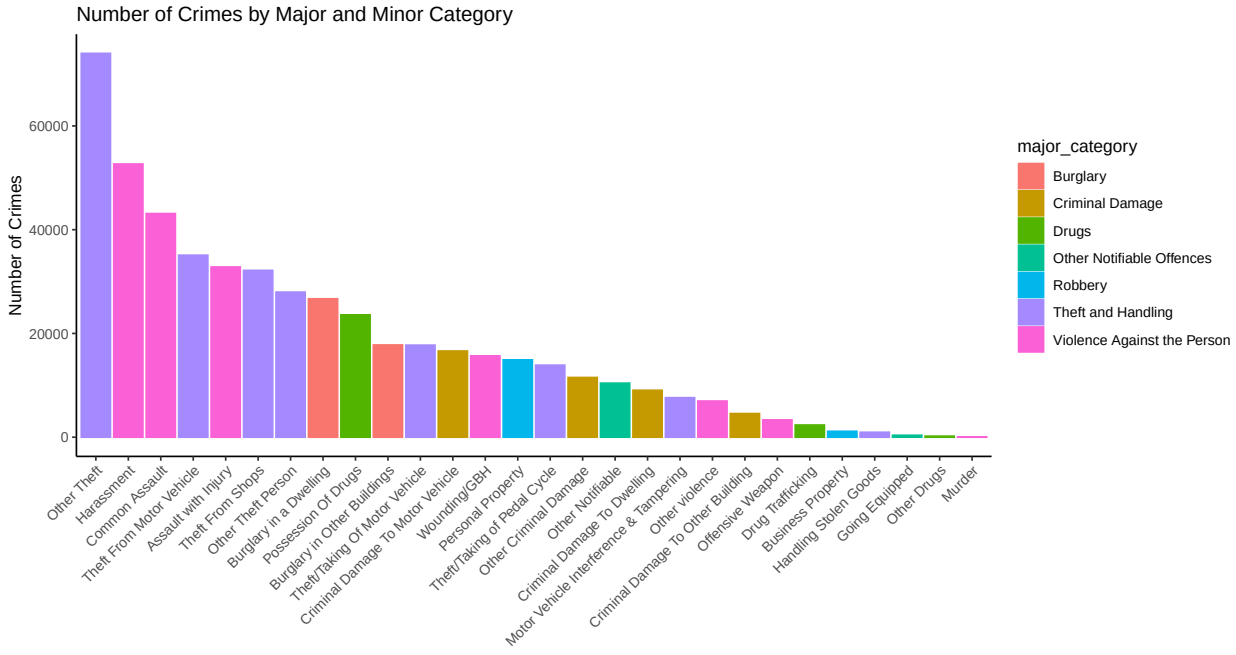
By far the highest number of crimes belong to the categories Theft and Handling as well as Violence Against the Person. There are no crimes documented with major category Fraud or Forgery and Sexual Offences.



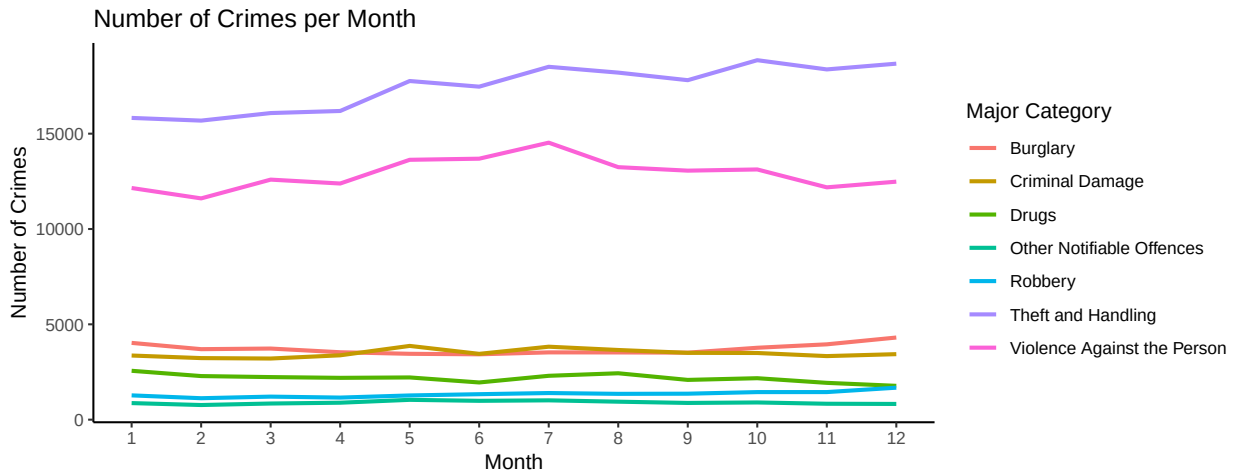

Other Theft is the minor category with the highest number of crimes, followed by Harassment and Common Assault. The latter two belong to the major category Violence Against the Person.



Combining the two categories, we can see that the seven top minor categories all belong to the major categories Theft and Handling and Violence Against the Person. Only in eighth place follows the minor category Burglary in a Dwelling which belongs to major category Burlary. Next is Possession of Drugs which belongs to major category Drugs.

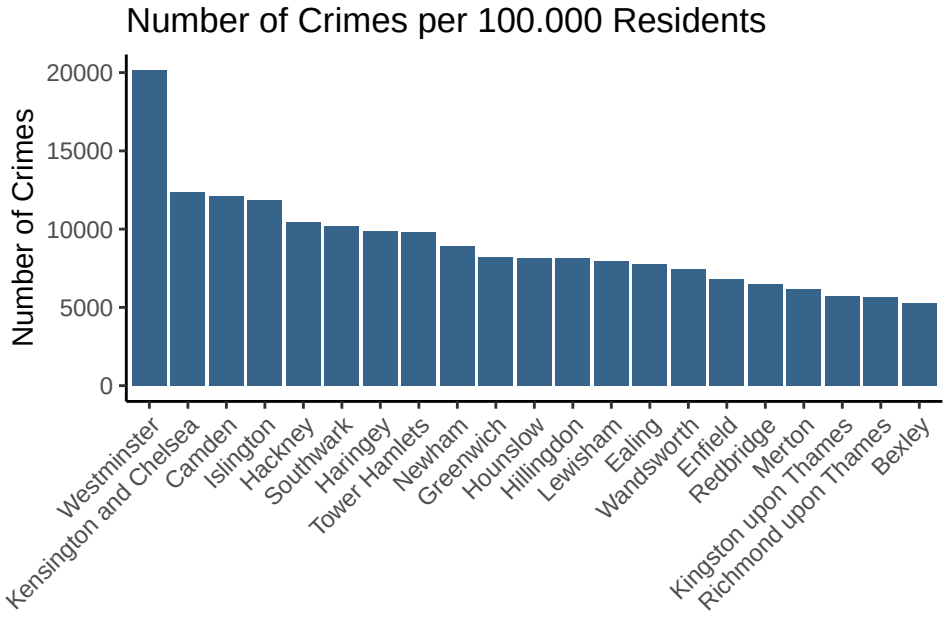



Especially Theft and Handling seems to be driving the overall trend of a rising number of crimes. Violence Against the Person also contributes a lot to the rising number of crimes during the middle of the year, but then decreases again. The other five categories are relatively stable over time, only Burglary and Criminal Damage have switched places during the course of the year.

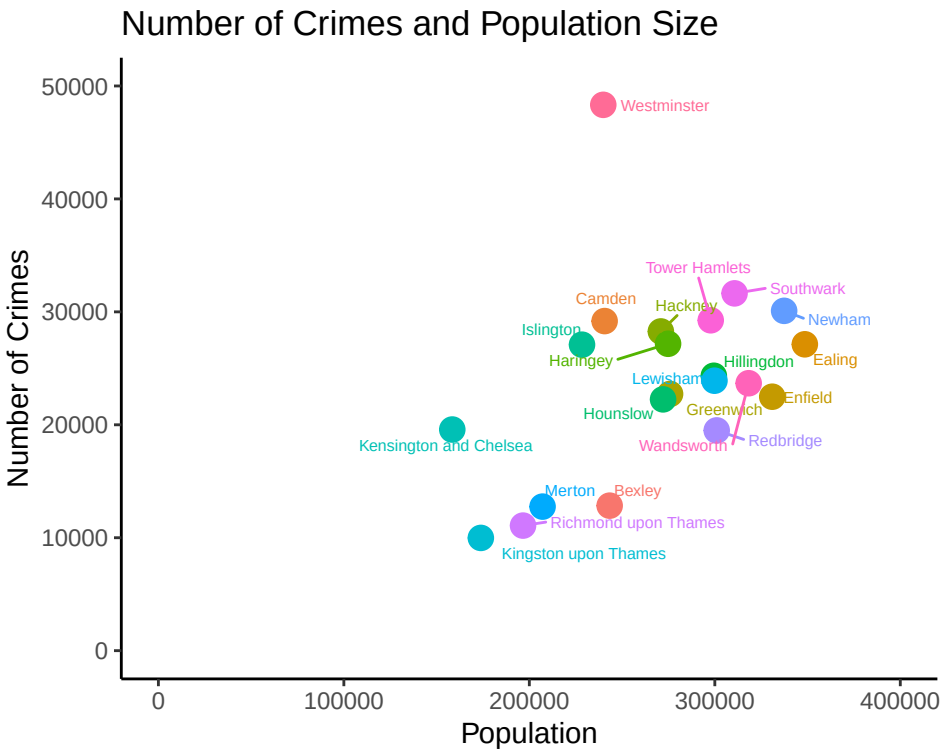


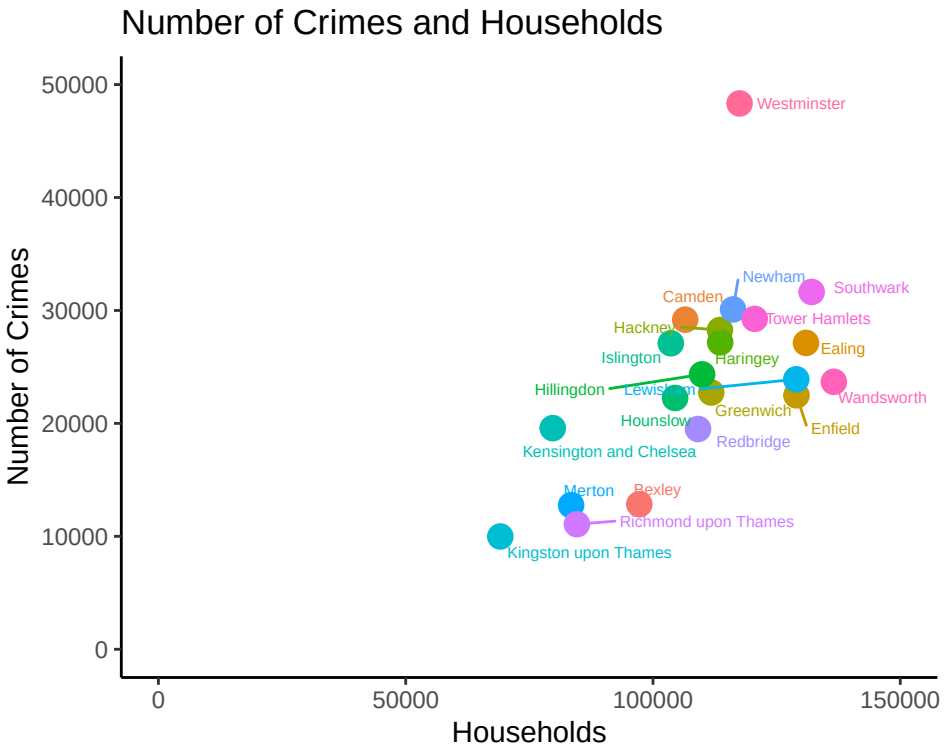
### 3.2.2 Population Size and Crimes

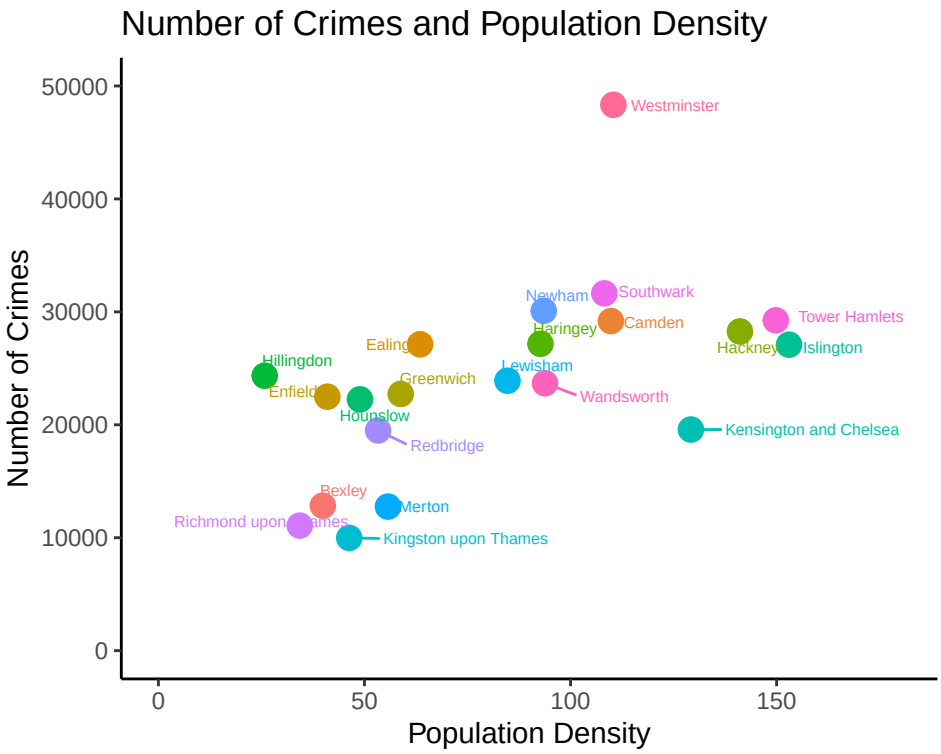
In this section we will investigate whether there is any correlation between the population variables and the number of crimes per borough. As the boroughs have a quite different size taking the absolute numbers, we will first compute the number of crimes per 100.000 residents to check whether there is a striking difference between the boroughs. However, the chart shows that the overall picture is still unchanged, by far the highest number of crimes is reported for Westminster. Then there is a slowly decreasing number over the other boroughs.



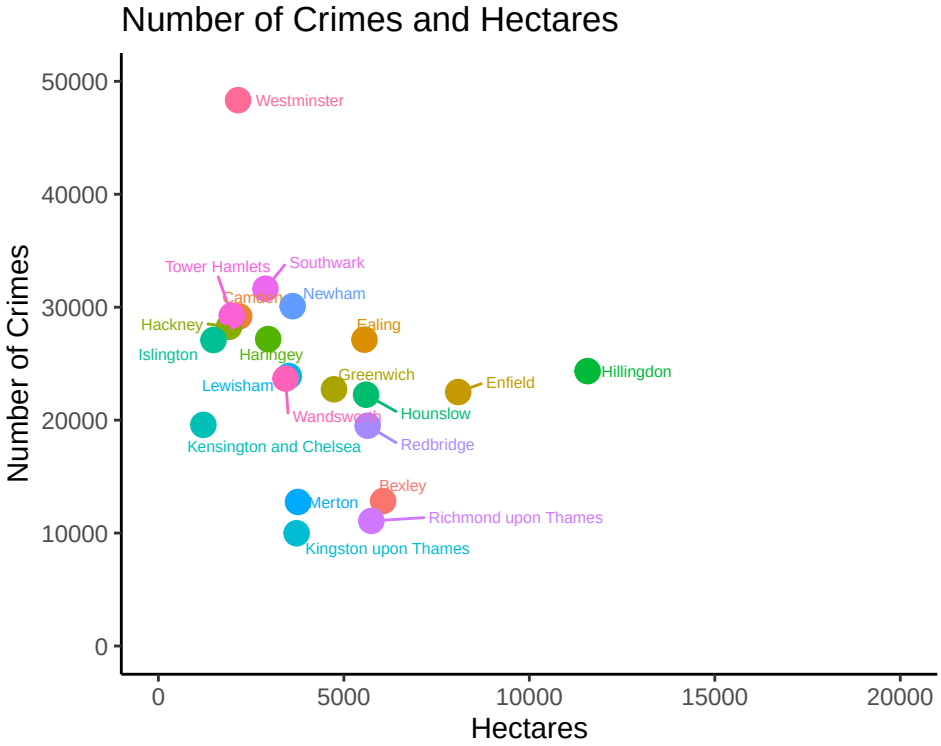

The scatterplot of crimes versus population size reveals that highly populated boroughs have a higher number of documented crimes. The same applies to crimes versus number of households in each borough. This correlation seems even clearer between crimes and population density. Logically, this also makes sense, as a densly populated area should offer both more potential victims and offenders, more opportunities to commit a crime and a highler potential for conflicts due to the crowdedness.


Number of Crimes and Population Size
Number of Crimes
Population
50000
40000
30000
20000
10000
0
0
100000
200000
300000
400000
Westminster
Tower Hamlets
Southwark
Camden
Hackney
Newham
Islington
Haringey
Hillingdon
Ealing
Lewisham
Enfield
Hounslow
Greenwich
Kensington and Chelsea
Wandsworth
Redbridge
Merton
Bexley
Richmond upon Thames
Kingston upon Thames


Number of Crimes and Households
Number of Crimes
Households
50000
40000
30000
20000
10000
0
0
50000
100000
150000
Westminster
Newham
Southwark
Camden
Tower Hamlets
Hackney
Ealing
Islington
Haringey
Hillingdon
Lewisham
Wandsworth
Greenwich
Enfield
Hounslow
Redbridge
Kensington and Chelsea
Merton
Bexley
Richmond upon Thames
Kingston upon Thames




In contrast, hectares seems like a less promising indicator for high number of crimes compared to population, households or population density. Hillingdon, for example, has about the same number of crimes as Islington but they have completely different hectares.



**Correlation Coefficient**

We now have three variables (population, household, population density) that we could use as predictors for the number of crimes, but logically they should all be correlated with each other. Thus, we don't want to include all three of them in a prediction model, as all kind of measure the same issue. We just want to include the strongest predictor of them, to keep our model nice and concise, instead of bloating it with all variables we can get our hands on. To find out which of the three has the strongest relationship with number of crimes, we are going to compute the correlation coefficient. The correlation coefficient $r$ measures the strength and direction of a linear relationship between two variables on a scatterplot. The value of $r$ is always between +1 and –1, where 0 means no relationship, +1 is a perfect positive relationship and -1 translates to a perfect negative relationship (source: https://en.wikipedia.org/wiki/Pearson_correlation_coefficient, accessed on 05/01/2020). We've used the R function `cor()` with the default method here, which is the Pearson Correlation Coefficient, according to the R Documentation (source: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/cor, accessed on 05/01/2020).

| cor(population, household) | cor(population, population_density) | cor(population, hectares) |
|---:|---:|---:|
| 0.8926454 | -0.0560188 | 0.3298699 |

| cor(household, population_density) | cor(household, hectares) | cor(population_density, hectares) |
|---:|---:|---:|
| 0.2138107 | 0.0648681 | -0.8274973 |

Between the population variables, there are the following correlations, based on the correlation coefficient calculated with the train dataset:

- high positive correlation between population and households
- very weak negative correlation between population and population density
- medium positive correlation between population and hectares
- weak positive correlation between households and population density
- medium positive correlation between households and hectares
- strong negative correlation between population density and hectares

Next, we compute the correlation coefficient for the population variables versus number of crimes. We can read from the table below that the strongest correlation is crimes versus households, followed by crimes and population density. Same as in the scatterplot, crimes and hectares seem to have only a weak negative correlation based on the correlation coefficient. Population and households are very strongly correlated with each other, so we will just include one of them in the prediction model, which is households as the correlation to crimes is the strongest.

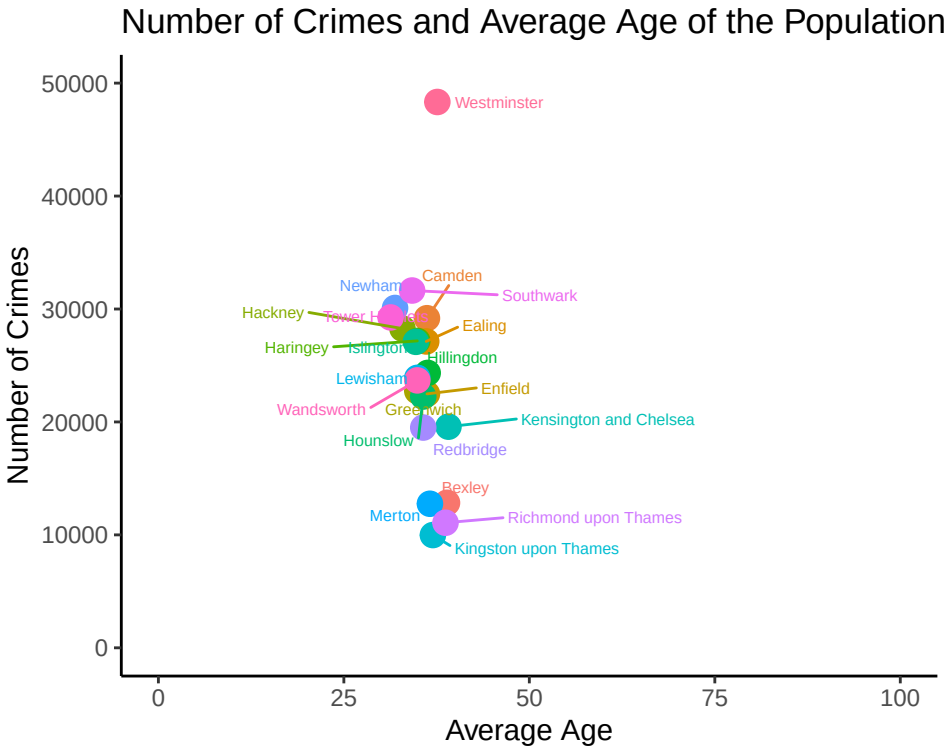
| cor(crimes, population) | cor(crimes, household) | cor(crimes, population_density) | cor(crimes, hectares) |
|---:|---:|---:|---:|
| 0.4019146 | 0.6212407 | 0.5663464 | -0.314878 |

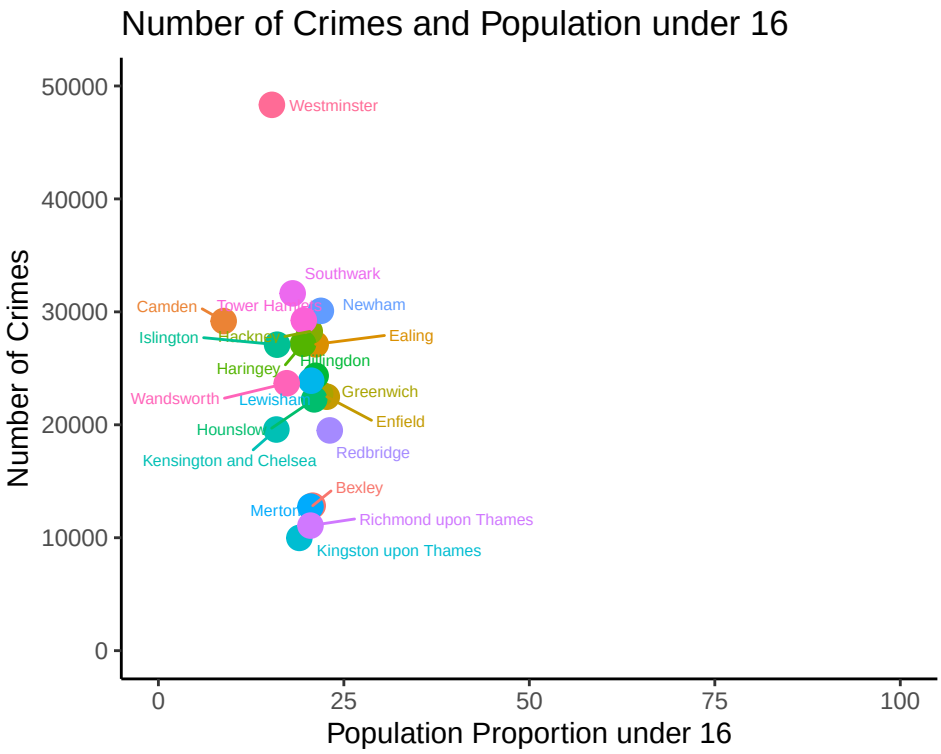
### 3.2.3 Population Age and Crimes

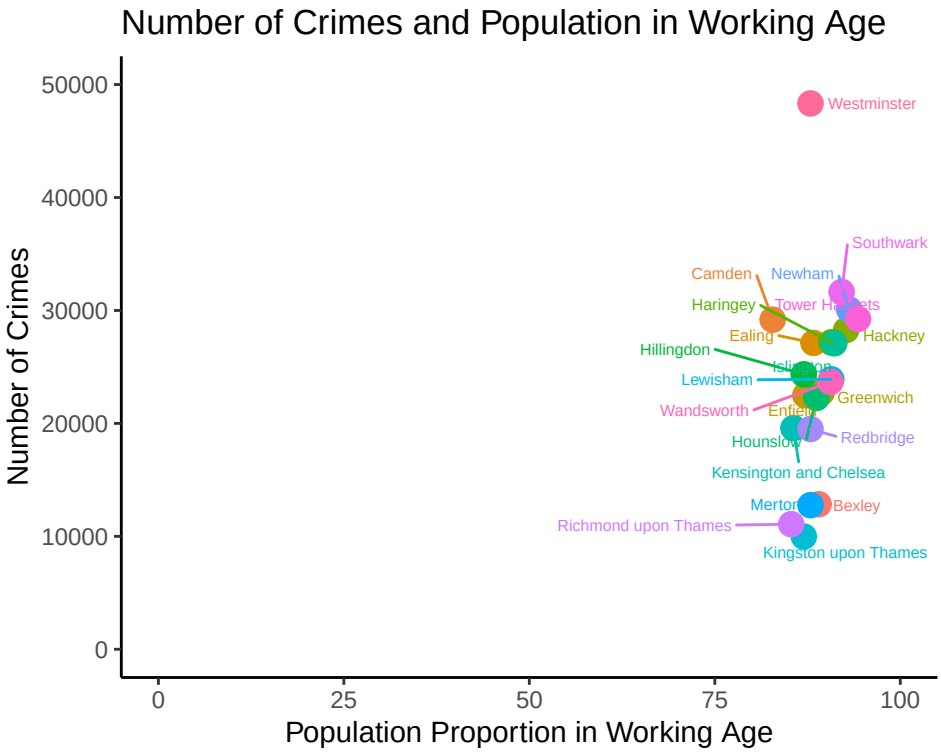
Similar to the variables about population size, we are now going to analyze the four variables revolving about population age. Why could population age play a role for the number of crimes? Again this is a matter of potential offenders and victims as well as opportunities for crime. Young children and elderly people are

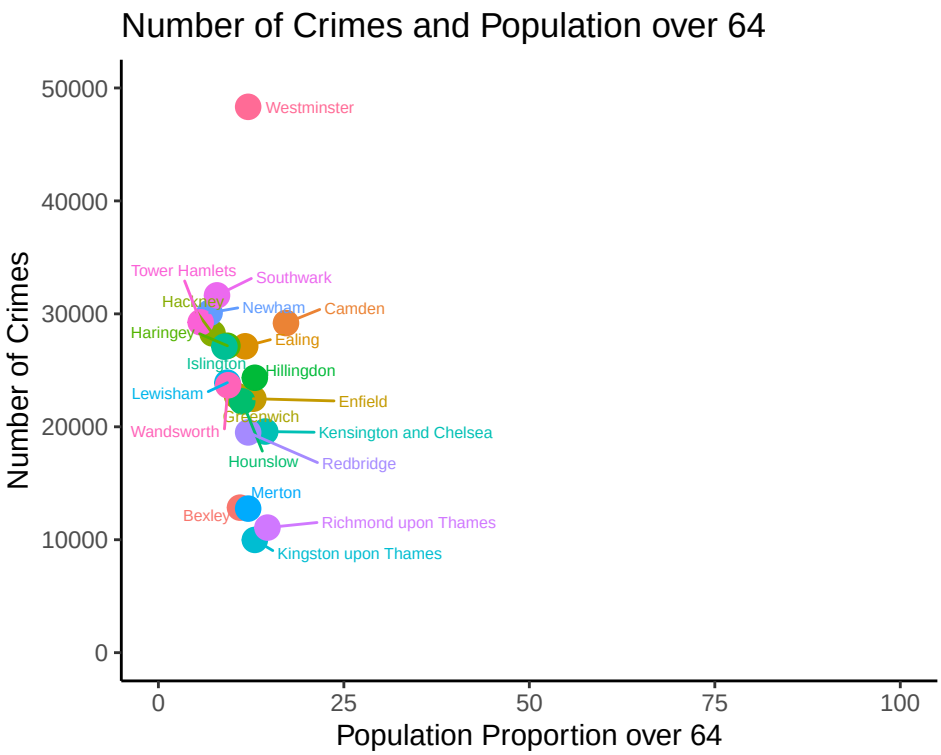
very unlikely to commit crimes. However, especially elderly people are often quite vulnerable for crimes and e.g. fall victim to theft.

The charts below illustrate that there could be a weak negative correlation between average age and crimes. The same applies to crimes and population proportion under 16 as well as population proportion over 64. So generally speaking, a high proportion of children and teenagers or elderly people suggest a borough with a lower number of crimes. On the contrary, population proportion in working age seems to be positively correlated with crimes. In all cases, Westminster seems to be somewhat of an outlier again. Overall, the population age seems to have only little effect on the total number of crimes, as all correlations appear to be rather weak.




Number of Crimes and Population under 16
Number of Crimes
50000
40000
30000
20000
10000
0
Westminster
Southwark
Camden
Tower Hamlets
Newham
Islington
Hackney
Ealing
Hillingdon
Haringey
Wandsworth
Lewisham
Greenwich
Enfield
Hounslow
Redbridge
Kensington and Chelsea
Bexley
Merton
Richmond upon Thames
Kingston upon Thames
0
25
50
75
100
Population Proportion under 16


Number of Crimes and Population in Working Age
Number of Crimes
50000
40000
30000
20000
10000
0
Westminster
Southwark
Camden
Newham
Haringey
Tower Hamlets
Ealing
Hackney
Hillingdon
Islington
Lewisham
Greenwich
Wandsworth
Enfield
Redbridge
Hounslow
Kensington and Chelsea
Merton
Bexley
Richmond upon Thames
Kingston upon Thames
0
25
50
75
100
Population Proportion in Working Age



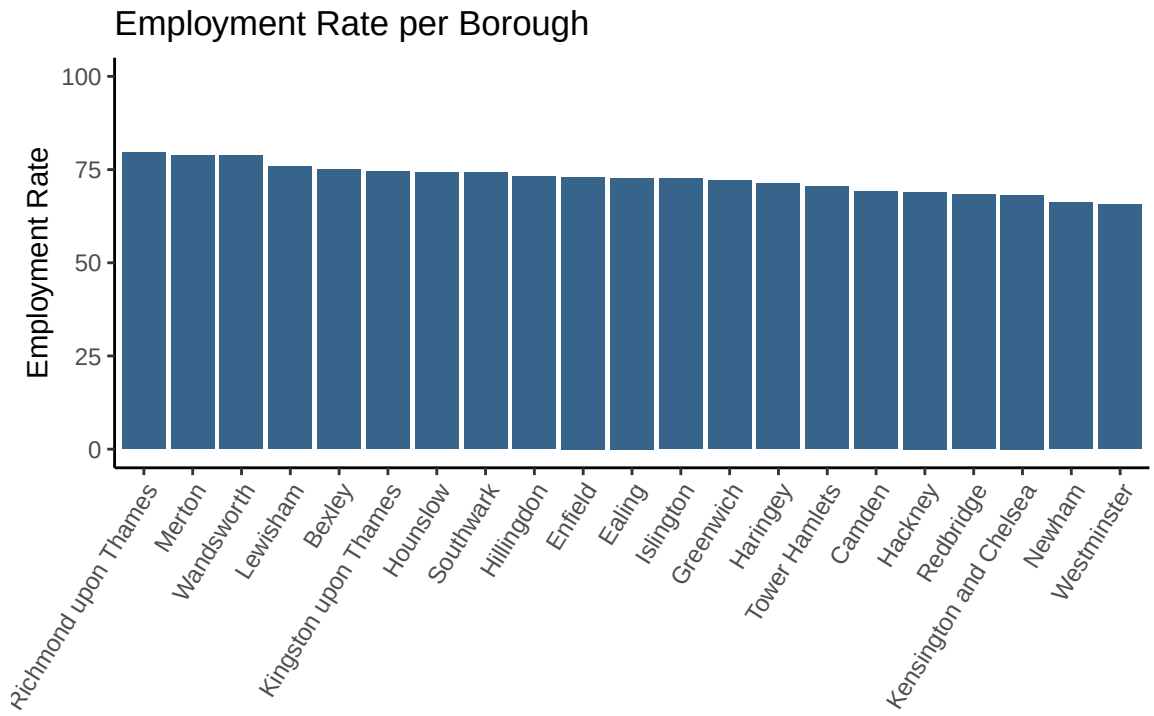

The correlation coefficients confirm our findings from the scatterplots. The relationship between crimes and average age seems to be the strongest.

| cor(crimes, avg_age) | cor(crimes, population_under_16) | cor(crimes, population_working_age) | cor(crimes, population_over_64) |
|---|---|---|---|
| -0.4044499 | -0.3541242 | 0.3166128 | -0.3166128 |

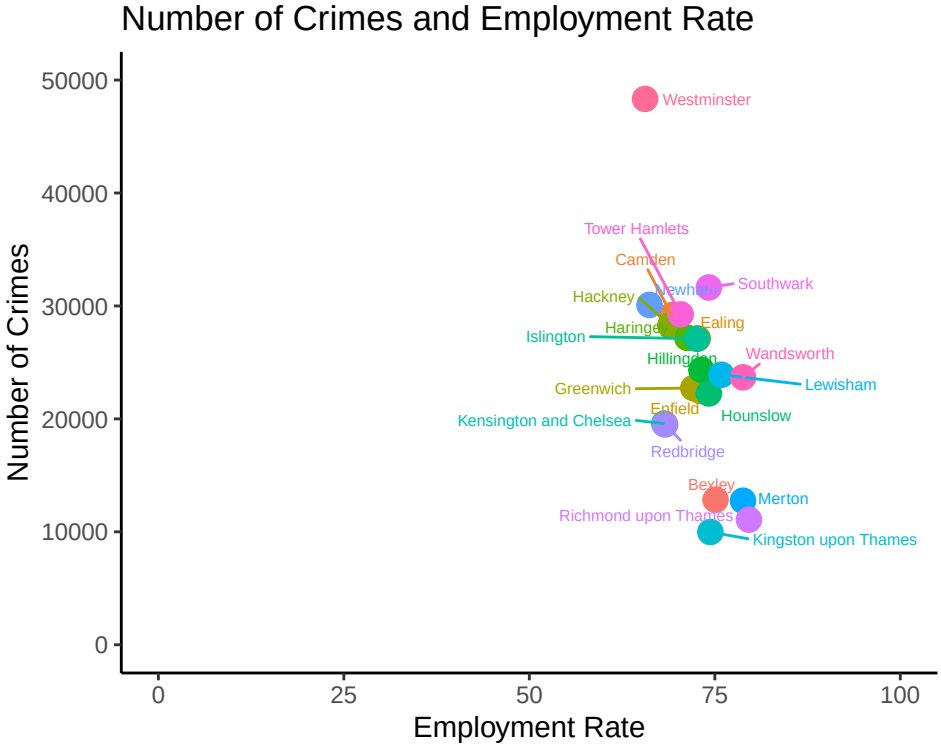
### 3.2.4 Employment and Crimes

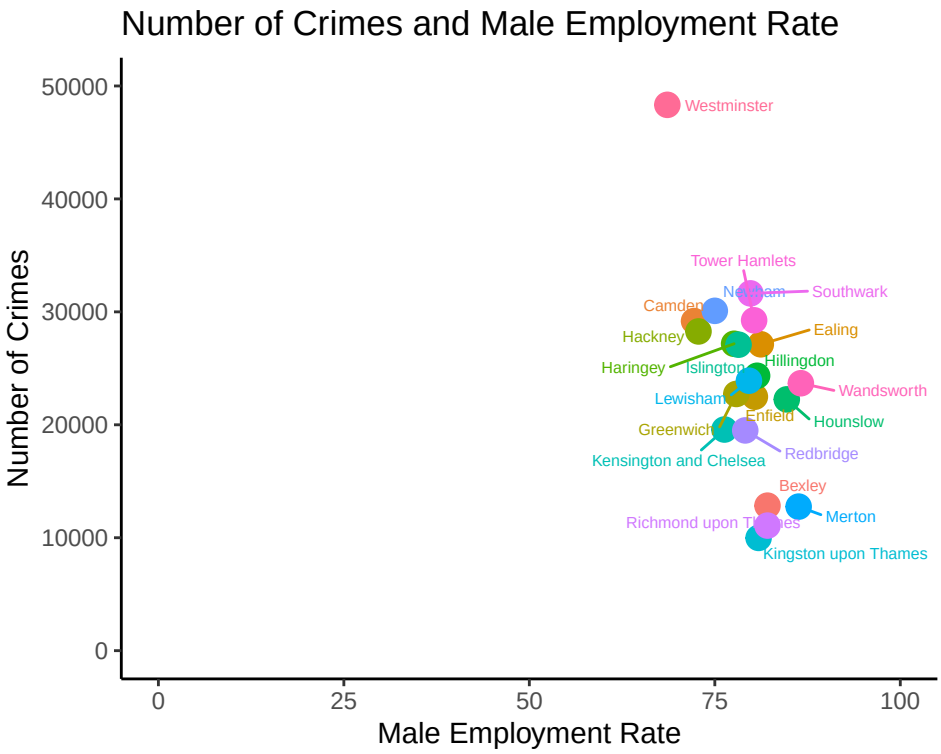
The reason to include a variable about employment or unemployment in a prediction model for crimes is pretty straightforward. Naturally, crimes are higher in a period of high unemployment as people are less likely to be able to fulfill their needs and earn money through a legal occupation. Furthermore unemployment drives insecurity, instability and conflict potential, which all foster crimes.

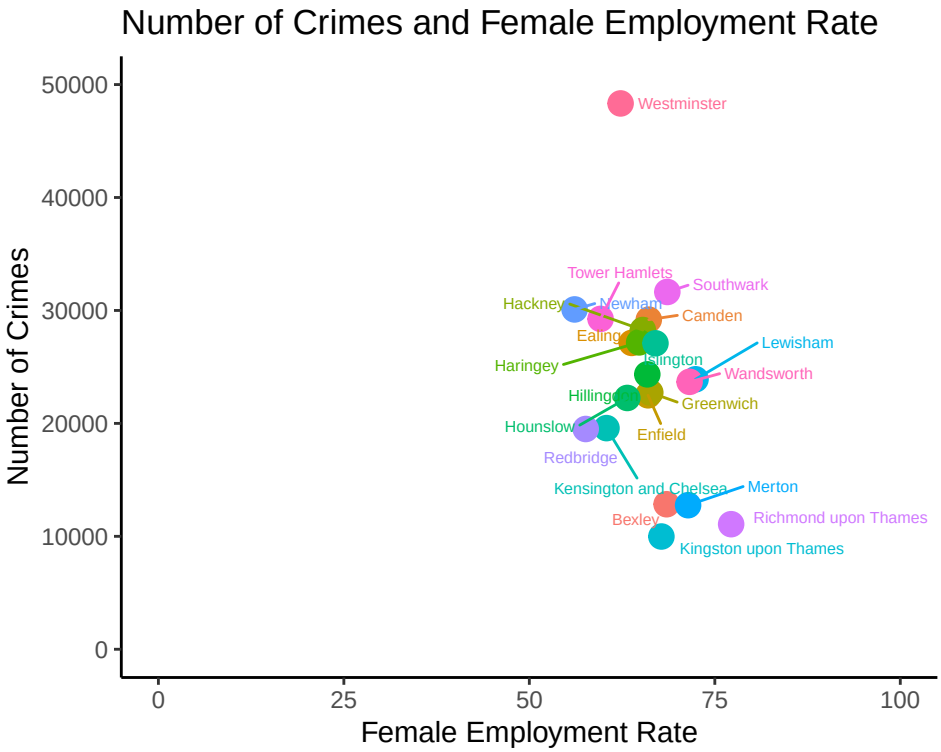
The bar chart of employment rates per borough reveals that the employment rates seem to be relatively high in all boroughs. Interestingly, Westminster which was the borough with the highest number of crimes, has the lowest employment rate. Suburban boroughs like Richmond upon Thames and Merton have the highest employment rates and are among the boroughs with the least crimes.

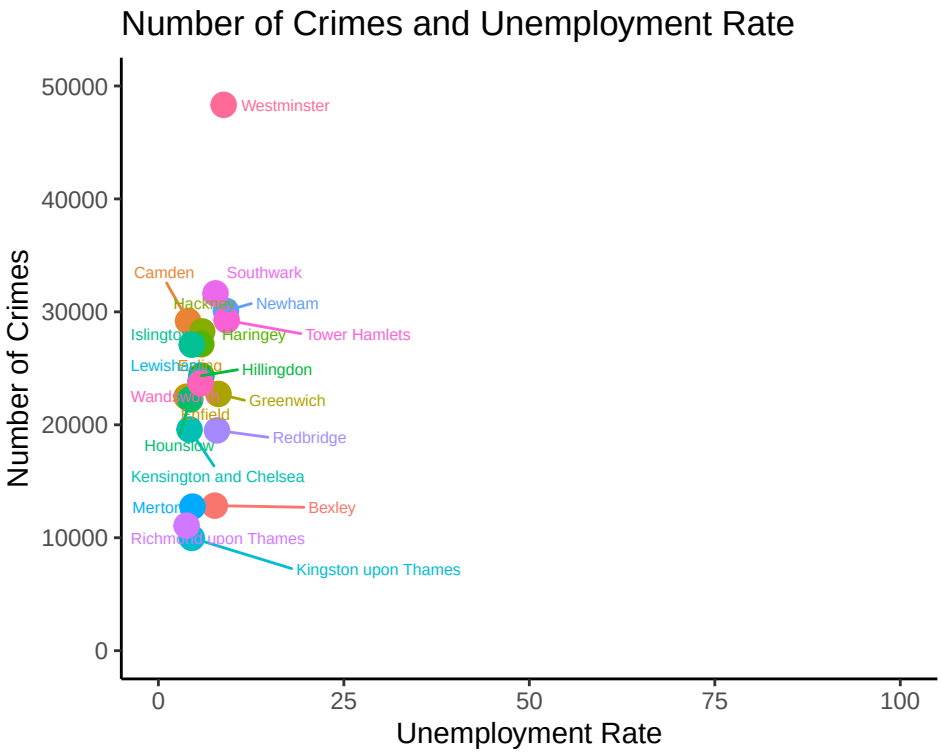



The scatterplots for crimes versus the four employment variables show that there is somewhat of a negative correlation between crimes and employment. The relationship between male employment rate and crimes appears to be the strongest, whereas the relationship between female employment rate and crimes appears to be the weakest.




Number of Crimes and Male Employment Rate
Number of Crimes
50000
40000
30000
20000
10000
0
0
25
50
75
100
Male Employment Rate
Westminster
Tower Hamlets
Southwark
Camden
Hackney
Ealing
Haringey
Hillingdon
Wandsworth
Lewisham
Hounslow
Greenwich
Enfield
Redbridge
Kensington and Chelsea
Bexley
Merton
Kingston upon Thames


Number of Crimes and Female Employment Rate
Number of Crimes
50000
40000
30000
20000
10000
0
0
25
50
75
100
Female Employment Rate
Westminster
Tower Hamlets
Southwark
Hackney
Camden
Lewisham
Haringey
Wandsworth
Greenwich
Hounslow
Enfield
Redbridge
Kensington and Chelsea
Merton
Bexley
Richmond upon Thames
Kingston upon Thames




Due to the fact that the employment rates male and female are missing from the dataset for the borough City of London and we have only formed an assumption for the employment rate male, the correlation coefficients shown below are excluding the borough City of London. Still, the findings from the scatterplots are again confirmed by the correlation coefficient which shows the strongest correlation between crimes and employment rate male.

| cor(crimes, employment_rate_2015) | cor(crimes, employment_rate_male_2015) |
|---|---|
| -0.6326631 | -0.6717671 |

| cor(crimes, employment_rate_female_2015) | cor(crimes, unemployment_rate_2015) |
|---|---|
| -0.4270386 | 0.4792179 |

**Results of the Exploratory Data Analysis**

The number of crimes varies depending on months and is also influenced by the population living in the respective area. The most promising predictors to include in our prediction model seem to be months, households, population density as well as average age and male employment rate.

# 4 Prediction Model

This chapter is devoted to the development of the prediction model. During the analysis we have found out that regression should be a suitable method to use and we have identified the most important predictors in our dataset. Before we can actually start with different models, we have to define the metric we want to evaluate our models with.

## 4.1 Metric to Evaluate the Model: RMSE

Generally, there are various different metrics one can use to evaluate model performance. Very popular metrics are for example

- R-Squared ($R^2$)
- Adjusted R-Squared (Adj $R^2$)
- Mean Square Errors (MSE)
- Root Mean Squared Errors (RMSE).

There is no metric which is generally the best for any prediction problem. It highly depends on the data you are using as well as the prediction problem you want to solve. In the Machine Learing course (PH125.8x), for example, we were faced with a categorical outcome and intially used overall accuracy to evaluate our model, which is one of the simplest evaluation methods. Later we have moved on to studying sensitivity and specificity. "Sensitivity, also known as the true positive rate or recall, is the proportion of actual positive outcomes correctly identified as such. Specificity, also known as the true negative rate, is the proportion of actual negative outcomes that are correctly identified as such" (source: PH125.8x Machine Learning).

In our case, we have a continuous outcome, and RMSE is a suitable evaluation metric, as it tells us how close our predicted regression line is to the actual data points. Furthermore, RMSE is relatively easy to interpret, because it comes in the same unit as the predicted variable (source: https://towardsdatascience.com/predictive-modellers-guide-to-choosing-the-best-fit-regression-model-707120e502b4, accessed on 26/04/2020). So for the problem at hand an RMSE of 500 would mean that our prediction misses by 500 crimes per month.

Assuming that $\hat{y}_i$ are our predicted values, $y_i$ are the observed values and $n$ is the number of observations, we can write the formula for the RMSE as follows (source: https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e, accessed on 05/16/2020):

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

## 4.2 Base Prediction: Average

Our base prediction will be the monthly average. This means we would simply assume that the number of crimes only varies based on month, but other than that is the same for each borough, and all other variation is just random variation. The monthly average is given in the following table:

| month | avg_crimes |
|---|---|
| 1 | 1908 |
| 2 | 1828 |
| 3 | 1900 |
| 4 | 1892 |
| 5 | 2059 |
| 6 | 2015 |
| 7 | 2148 |
| 8 | 2064 |
| 9 | 2010 |
| 10 | 2084 |
| 11 | 2002 |
| 12 | 2056 |

Using this prediction, we get an RMSE of 738 on the test set. This means we are missing by over 700 crimes per month. Considering that Westminster was the borough with the higest number of crimes ranging around 4.000 per month and many boroughs had around 1.000 crimes per month, the RMSE seems quite high. We should certainly be able to beat this benchmark with more sophistcated prediction methods. Just for fun I've also added a table with the output of the R function `accuracy()`, in order to check out other evaluation metrics as well. The table shows the RMSE, but also ME (Mean Error), MAE (Mean Absolute Error), MPE (Mean Percentage Error) and MAPE (Mean Absolute Percentage Error) (source: https://www.rdocumentation.org/packages/forecast/versions/8.12/topics/accuracy, accessed on 05/16/2020).

| Method | RMSE |
|---|---|
| Average Crimes per Borough | 738 |

| | ME | RMSE | MAE | MPE | MAPE |
|---|---|---|---|---|---|
| Test set | -336 | 738 | 642 | -40 | 51 |

## 4.3 Linear Regression

The second prediction method we will try is Linear Regression. During the EDA we have found five variables that we could potentially include in our prediction model. As we don't really know yet how much each of them improves our prediction results, we will first start with the simplest version only including month and then subsequently add the other four variables while checking the RMSE for each version of the LM model. The code below shows the implementation in R. The dataset `train_red` is a reduced version of the train dataset, which only includes the variables needed for the regression. The LM prediction with month only is almost the same as just the simple average, so the RMSE are also almost the same.

```
# fit regression line to predict crimes
fit_lm0 <- lm(crimes ~ month, data = train_red)

# check out the statistics of the fit
tidy(fit_lm0, conf.int = TRUE)
```

```
## # A tibble: 2 x 7
##   term        estimate std.error statistic  p.value conf.low conf.high
```

```
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>    <dbl>     <dbl>
## 1 (Intercept)   1878.       95.1     19.8  5.83e-53  1691.      2066.
## 2 month           18.3      12.9      1.41 1.59e- 1    -7.19      43.7
```

```
# predict crimes for the test data with the fitted regression
predicted_crimes_lm0 <- predict(fit_lm0, newdata = test_red)

# evaluate the prediction with the test set
rmse_lm0 <- RMSE(reported_crimes, predicted_crimes_lm0)
round(rmse_lm0)
```

```
## [1] 738
```

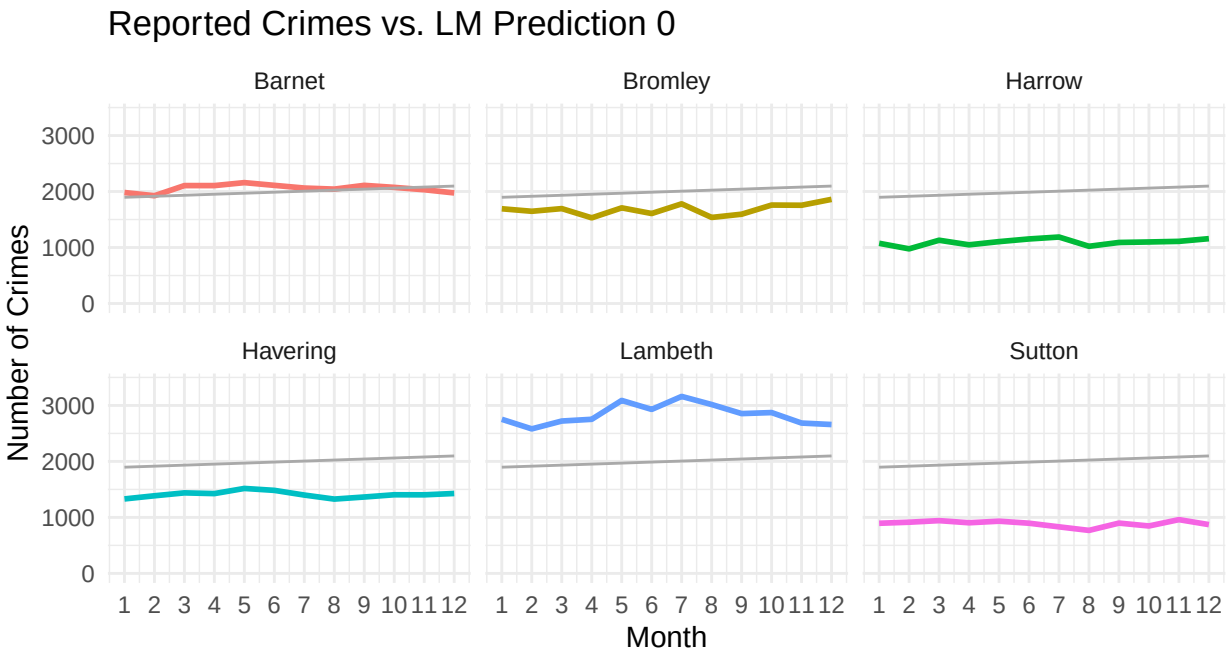
By adding one variable at a time to the Linear Regression, we get another four models. The resulting RMSE for each of them are shown below. The model with month, households and population density performs best.

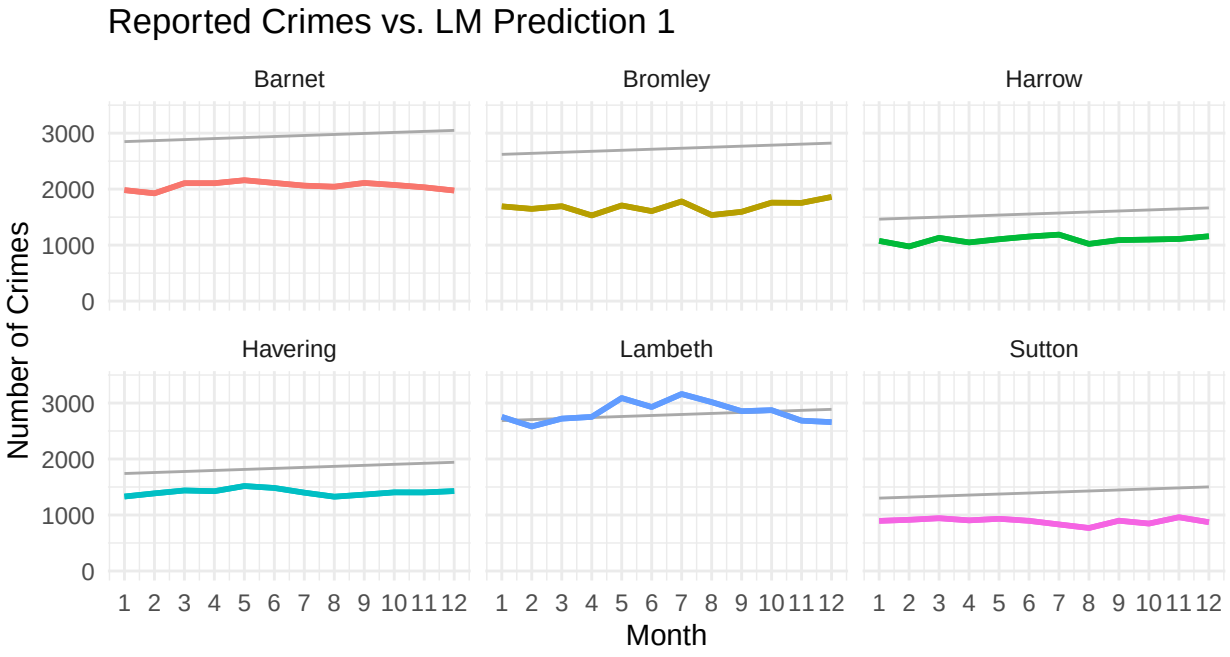
| Method | RMSE |
|---|---|
| Average Crimes per Borough | 738 |
| LM month | 738 |
| LM month+households | 661 |
| LM month+households+density | 325 |
| LM month+households+density+age | 467 |
| LM month+households+density+age+employment | 576 |

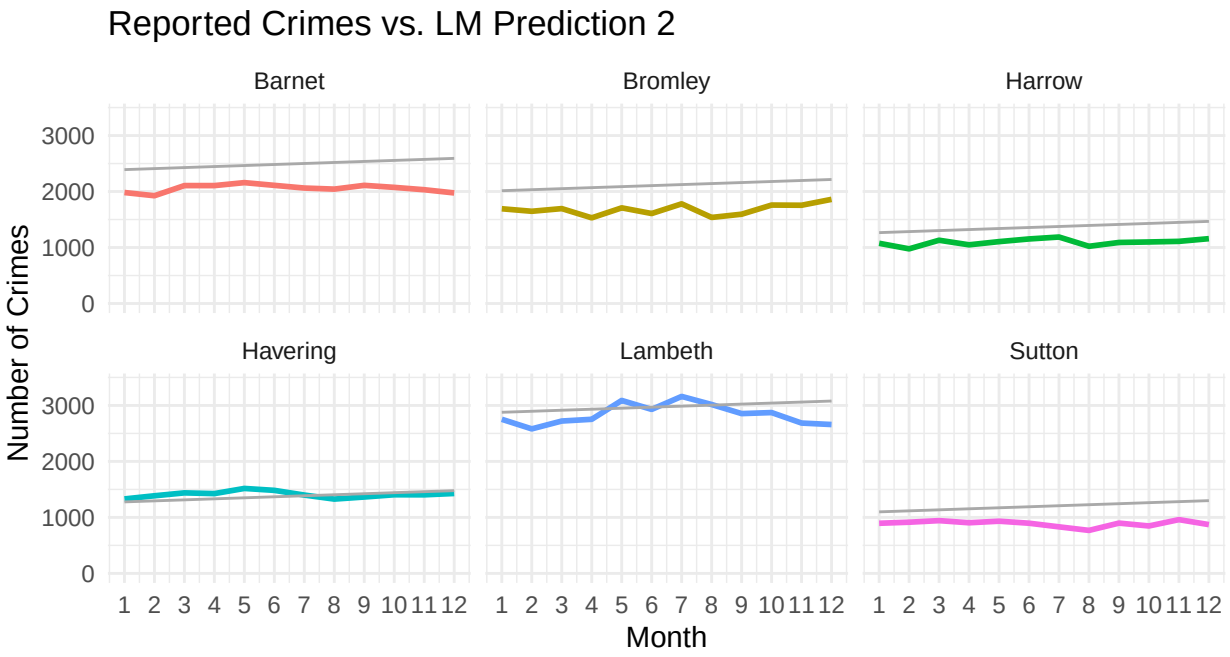
Besides studying the RMSE results it is always helpful to visualize the prediction versus the actual observations. With a visualization it is much easier to understand what the model is predicting well and what it is not predicting well. The following charts illustrate the predictions as grey line versus the actual observations for each borough in our test dataset.

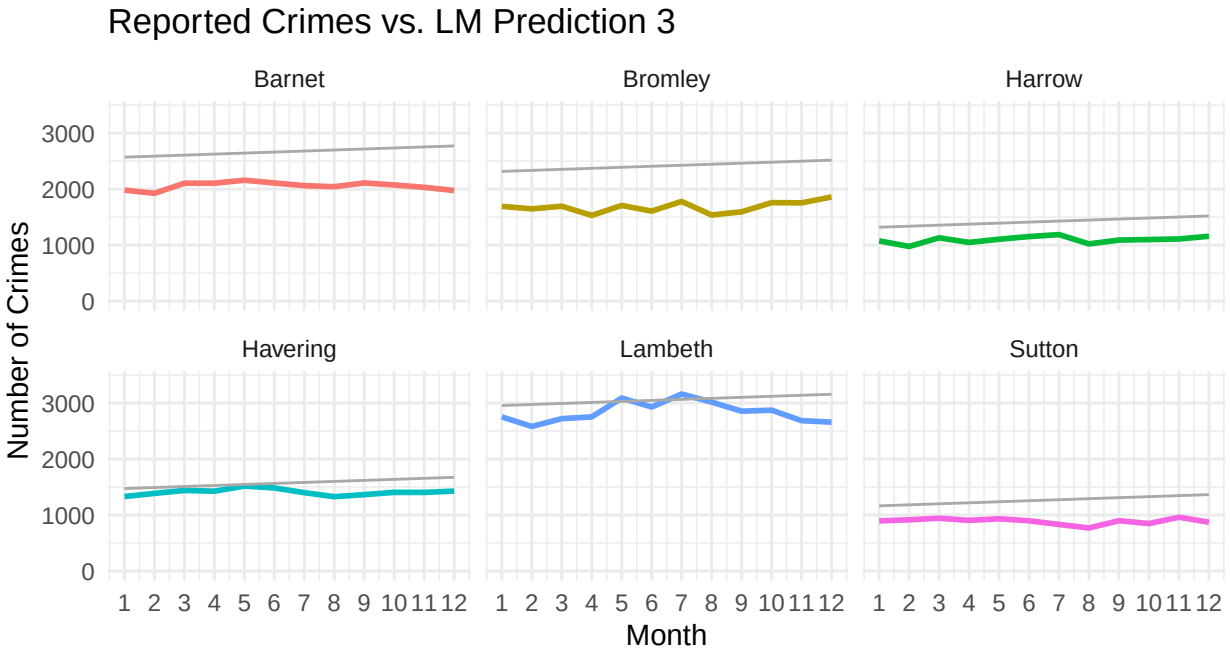
Prediction 0 is especially off for boroughs Harrow, Lambeth and Sutton, whereas prediction 1, 2 and 4 are predicting the number of crimes worst for Barnet and Bromley. Prediction 0 was the prediction just using month, and Barnet and Bromley are very close to the overall monthly average, that's why this prediction works well for these two boroughs. Harrow, Havering and Sutton are suburban boroughs with a relatively low number of crimes and the predictions for these three boroughs gain a lot by adding households, population density, average age and employment rate male to the model. Lambeth has the highest number of crimes in the test set as well as a high number of households and by far the highest population density, so the prediction gains most through these two variables. The same applies for Barnet and Bromley, which have a high number of households, but a rather low population density, thus the LM prediction 1 only with households is considerably overestimating the number of crimes, whereas the LM prediction 2 additionaly including population density is much closer to the actuals. Barnet and Bromley apparently are rather untypical considering average age and employment rate male, thus the two last predictions are performing worse than prediction 2.

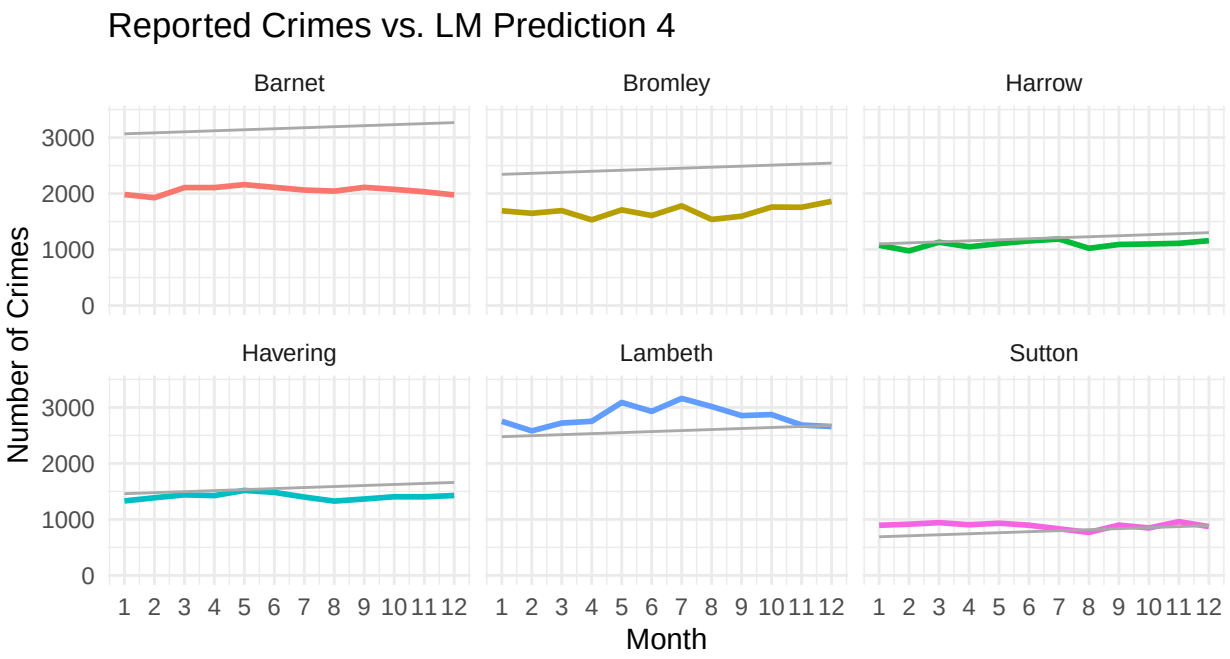
| Borough | AVG Crimes per Month | Households | Population Density | Average Age | Employment Rate Male |
|---|---|---|---|---|---|
| Barnet | 2057 | 149147 | 44.5 | 37.2 | 74.5 |
| Bromley | 1680 | 139654 | 21.7 | 40.1 | 80.4 |
| Harrow | 1096 | 91390 | 49.8 | 38.1 | 83.1 |
| Havering | 1409 | 102954 | 22.3 | 40.3 | 81.3 |
| Lambeth | 2839 | 142382 | 121.4 | 34.4 | 82.3 |
| Sutton | 888 | 84612 | 46.1 | 38.7 | 86.2 |


Reported Crimes vs. LM Prediction 0
Barnet
Bromley
Harrow
Havering
Lambeth
Sutton
Number of Crimes
0
1000
2000
3000
1 2 3 4 5 6 7 8 9 10 11 12
Month


Reported Crimes vs. LM Prediction 1
Barnet
Bromley
Harrow
Havering
Lambeth
Sutton
Number of Crimes
0
1000
2000
3000
1 2 3 4 5 6 7 8 9 10 11 12
Month


Reported Crimes vs. LM Prediction 2
Barnet
Bromley
Harrow
Havering
Lambeth
Sutton
Number of Crimes
0
1000
2000
3000
1 2 3 4 5 6 7 8 9 10 11 12
Month


Reported Crimes vs. LM Prediction 3
Barnet
Bromley
Harrow
Havering
Lambeth
Sutton
Number of Crimes
0
1000
2000
3000
1 2 3 4 5 6 7 8 9 10 11 12
Month




## 4.4 K-Nearest Neighbours Algorithm

The third method we are going to try is the K-Nearest Neighbours Algorithm. We will use the `train()` function in R, as we can simultaneously train and tune our model with this function. By default the `train()` function performs coss-validation by testing on 25 bootstrap samples comprised of 25% of the observations and also tries k = 5, 7 and 9 for `method = "knn"` (source: PH125.8x Machine Learning).

However, we don't just want to try these three values for k, we will set a different range of k's the function should try. When k = 1 the estimate for each $(x_1, x_2)$ in the training set is obtained with just the $y$ corresponding to that point. That would lead to very high accuracy in the train set, but also means we are over-training. On the other hand we can get over-smoothing if we select a k so large that we include very far away neighbours, which means we don't permit enough flexibility (source: PH125.8x Machine Learning). Our dataset is relatively small, so we won't get an issue with computational power and can try a larger range of k's, as long as we keep in mind that we want to prevent both over-trainig and over-smoothing. I've defined a sequence from 1 to 100 with increments of 1 to try out for tuning.

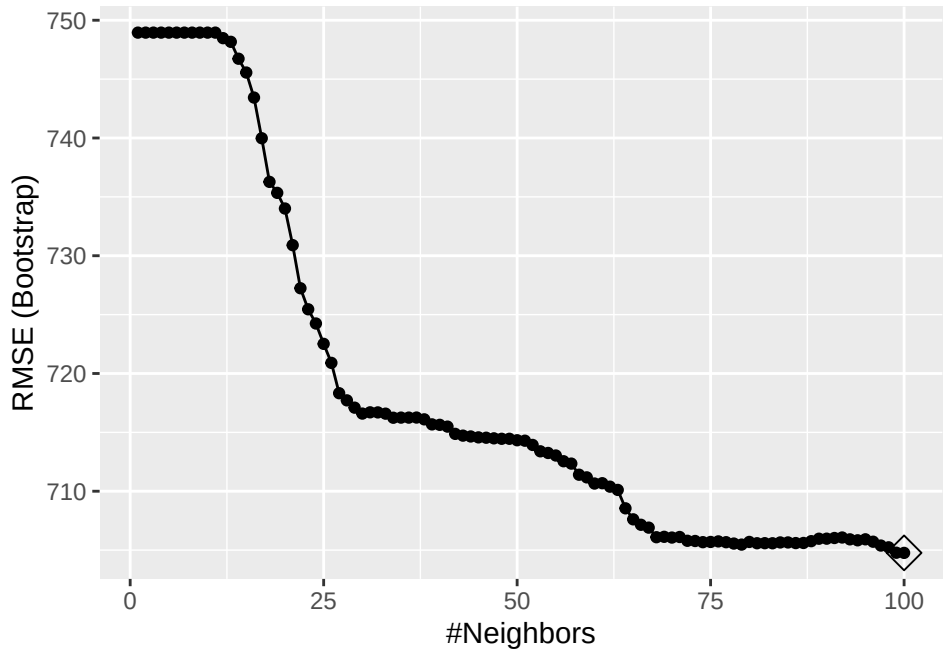
```
# fit knn to predict crimes and simultaneously tuning k
fit_knn0 <- train(crimes ~ month, method = "knn", data = train_red,
                  tuneGrid = data.frame(k = seq(1, 100, 1)))

# check which k is the best tune
fit_knn0$bestTune
```

```
##       k
## 100 100
```

```
fit_knn0$finalModel
```

```
## 100-nearest neighbor regression model
```

```
ggplot(fit_knn0, highlight = TRUE)
```



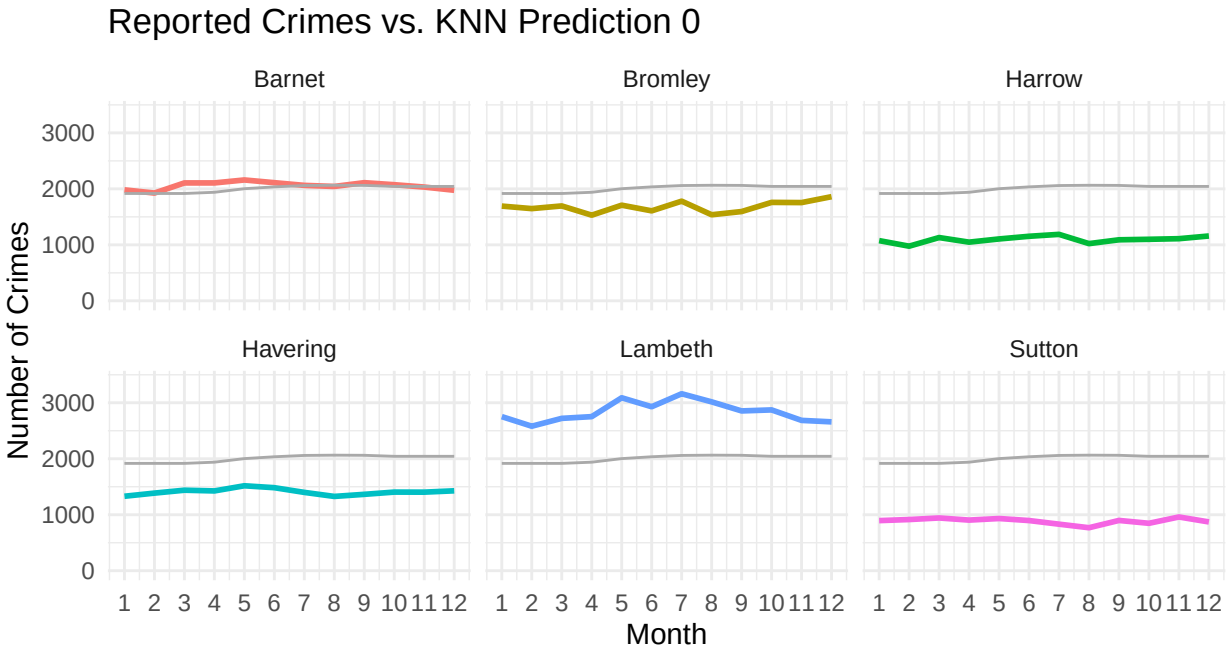

```
# predict crimes for the test data
predicted_crimes_knn0 <- predict(fit_knn0, test_red)

# evaluate the prediction
rmse_knn0 <- RMSE(reported_crimes, predicted_crimes_knn0)
```

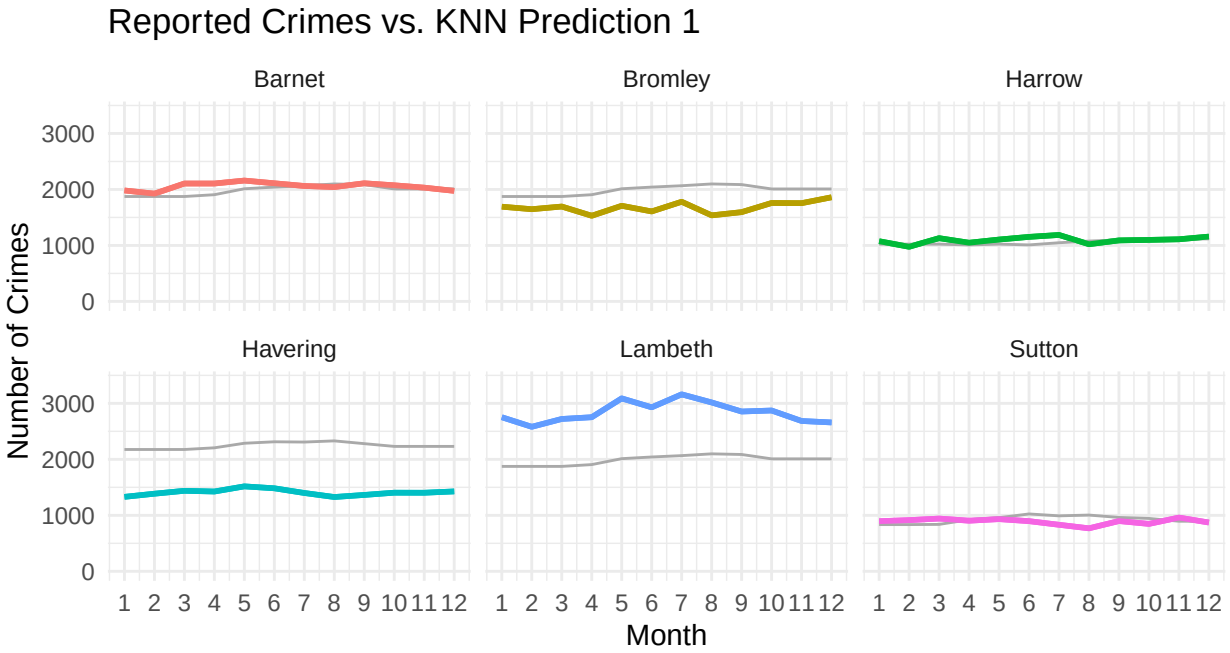
The chart above shows that the best tune for k is 100. However, the largest gains are achieved in the range between about 12 and 30, as the descent of the RMSE is the steepest in this area. Similar to the LM prediction, we also train the KNN-models in four additional versions subsequently adding household, population density, average age and employment rate male. The results are displayed in the table below. The KNN predictions with more than just month as regressor perform relatively well compared to our starting point of the overall average, but the LM model with month, households and population density is still by far the best model until now.

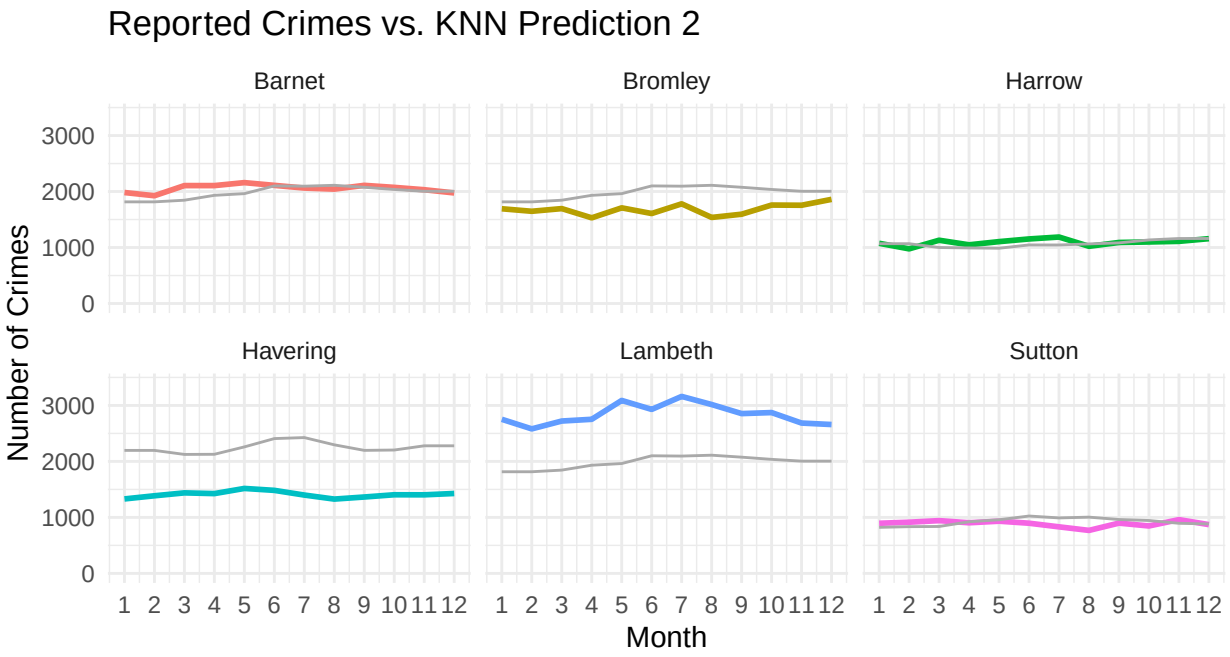
| Method | RMSE |
|---|---|
| Average Crimes per Borough | 738 |
| LM month | 738 |
| LM month+households | 661 |
| LM month+households+density | 325 |
| LM month+households+density+age | 467 |
| LM month+households+density+age+employment | 576 |
| KNN month | 740 |
| KNN month+households | 514 |
| KNN month+households+density | 519 |
| KNN month+households+density+age | 522 |
| KNN month+households+density+age+employment | 519 |

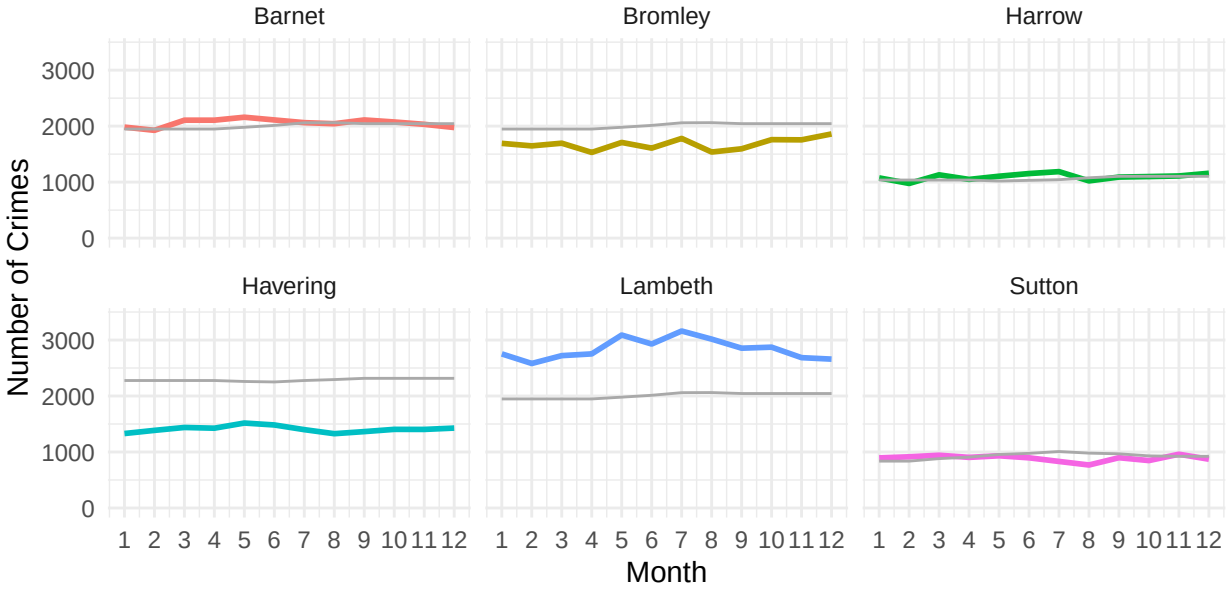
Looking at the charts below we can see that the KNN prediction is not a straight line like the LM prediction, but varies a bit during the year due to the weak seasonality that is given in each time series. Prediction 0 is worst for Harrow, Lambeth and Sutton, whereas the other four models all perform pretty well for Barnet,

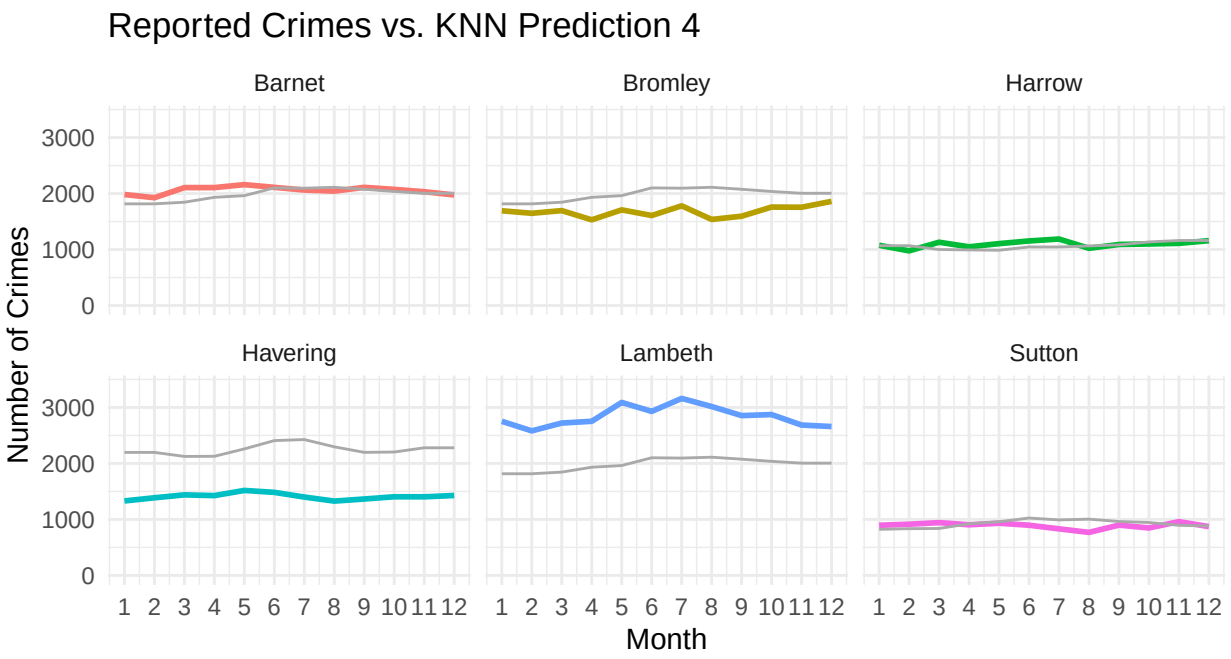
Bromley, Harrow and Sutton, but are relatively far off for Havering and Lambeth.




Reported Crimes vs. KNN Prediction 2
Barnet
Bromley
Harrow
Havering
Lambeth
Sutton
Number of Crimes
0
1000
2000
3000
1 2 3 4 5 6 7 8 9 10 11 12
Month


Reported Crimes vs. KNN Prediction 3
Barnet
Bromley
Harrow
Havering
Lambeth
Sutton
Number of Crimes
0
1000
2000
3000
1 2 3 4 5 6 7 8 9 10 11 12
Month

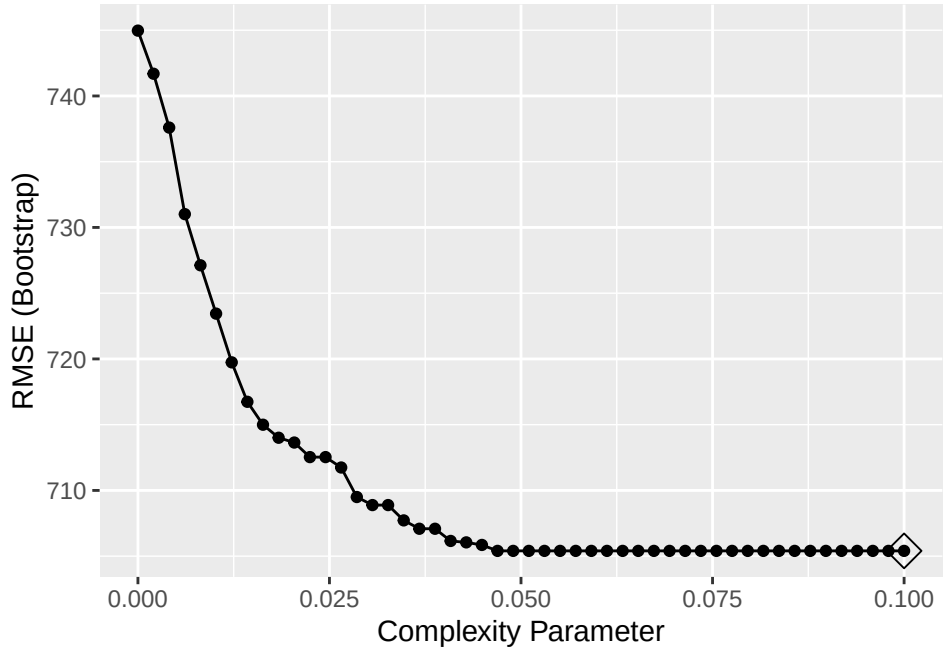



## 4.5 Regression Tree (rpart)

The last method we will try is a Regression Tree using the `rpart` package in R. Regression Trees or Decision Trees (for categorical outcomes) "build a decision tree and, at the end of each node, obtain a predictor $\hat{y}$. Mathematically, we are partitioning the predictor space into $J$ non-overlapping regions, $R_1, R_2, ..., R_J$ and then for any predictor $x$ that falls within region $R_j$, estimate $f(x)$ with the average of the training observations $y_i$ for which the associated predictor $x_i$ is also in $R_j$. To pick $j$ and its value $s$, we find the pair that minimizes the residual sum of squares (RSS)" (source: PH125.8x Machine Learning).

The `rpart` package defines two tuning parameters, cp which is the complexity parameter and minsplit which represents the minimum number of observations that are required in a partition in order to further partition it. The complexity parameter cp can be understood as the "minimum benefit" that an additional partition must add to the decision tree. `Rpart()` only partitions if the split adds at least this much benefit. If cp = 0 there are no restrictions to the paritioning, so the function will compute the most complex tree possible. In our case the dataset is relatively small and as stated above we shouldn't run into any issues due to a lack of computational power, so we can risk allowing the most complex tree possible. The selected range for the cp is 0 to 0.10.

```
# fit rpart to predict crimes and simultaneously tune cp
fit_rpart0 <- train(crimes ~ month,
                    method = "rpart", tuneGrid = data.frame(cp = seq(0, 0.10, len = 50)),
                    data = train_red)

# tuning: look at the RMSE for the different cp
ggplot(fit_rpart0, highlight = TRUE)
```

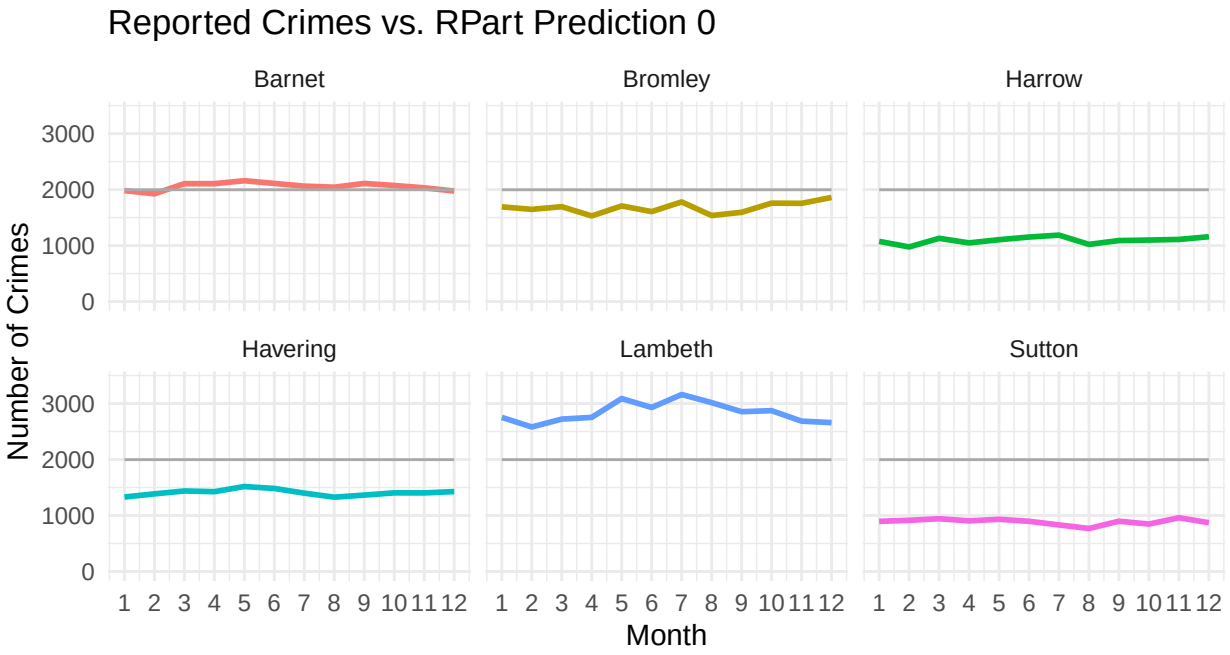


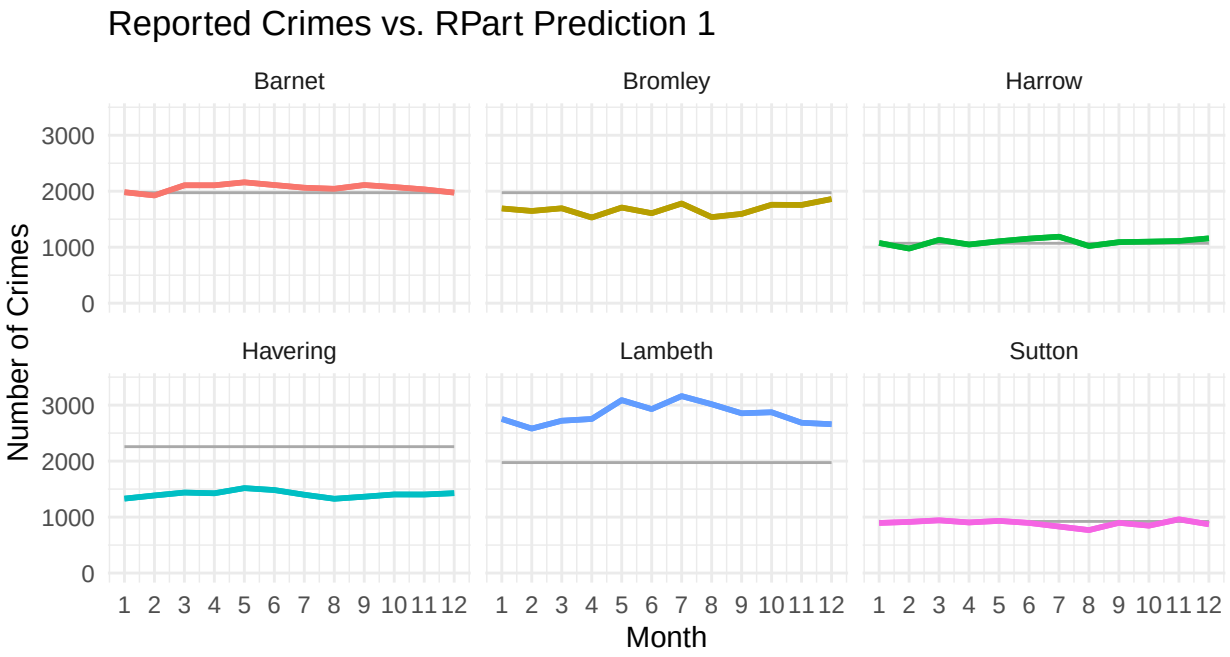

```
# predict crimes in the test set
predicted_crimes_rpart0 <- predict(fit_rpart0, newdata = test_red)

# evaluate prediction for the test set
rmse_rpart0 <- RMSE(reported_crimes, predicted_crimes_rpart0)
```

Again we train the other four model variations as well. The table below shows the results. Two of the Regression Tree models actually beat our best LM model. Due to the very good results of the Rpart prediction 2 and 4 I've also tried another version to see whether this further improves the prediction. The Rpart prediction 5 is a combination of rpart 2 and 4, using month + households + population density + employment rate male. Unfortunately the combined model performs worse than the single ones.

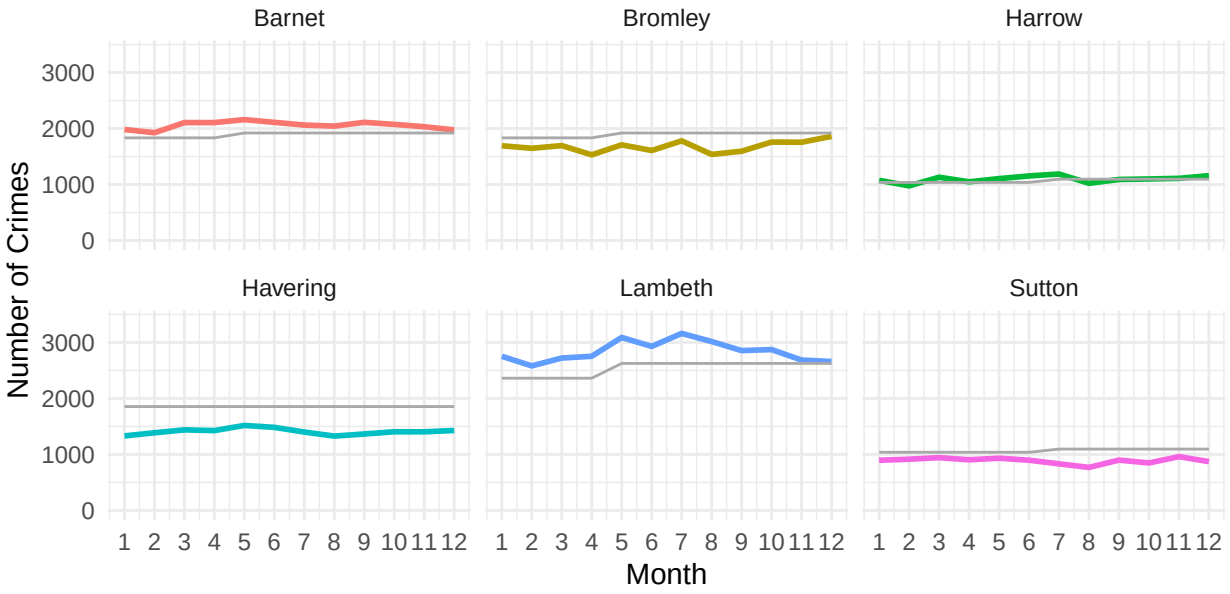
| Method | RMSE |
|---|---|
| Average Crimes per Borough | 738 |
| LM month | 738 |
| LM month+households | 661 |
| LM month+households+density | 325 |
| LM month+households+density+age | 467 |
| LM month+households+density+age+employment | 576 |
| KNN month | 740 |
| KNN month+households | 514 |
| KNN month+households+density | 519 |
| KNN month+households+density+age | 522 |
| KNN month+households+density+age+employment | 519 |
| RPart month | 736 |
| RPart month+households | 519 |
| RPart month+households+density | 271 |
| RPart month+households+density+age | 1654 |
| RPart month+households+density+age+employment | 285 |
| RPart month+households+density+employment | 430 |

The charts again reveal that prediction 0 is worst for Harrow, Lambeth and Sutton. Prediction 1, in contrast, is farthest off for Havering and Lambeth. Prediction 3 failed so much for Barnet, Bromley and Havering that we even needed to enlarge the y-axis. Prediction 2 and 4, however, are a pretty close fit.



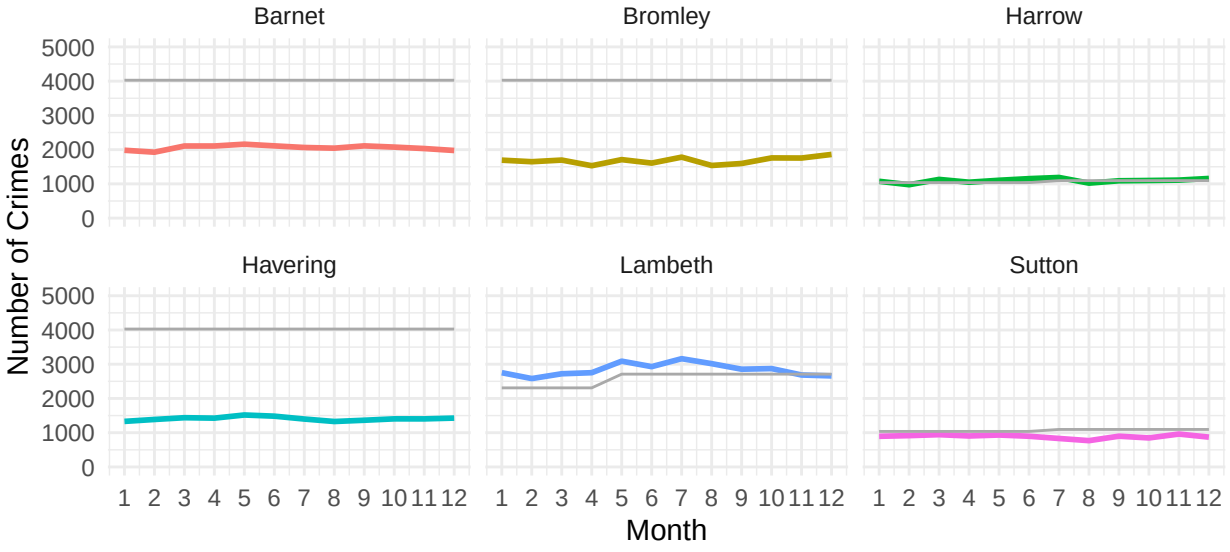



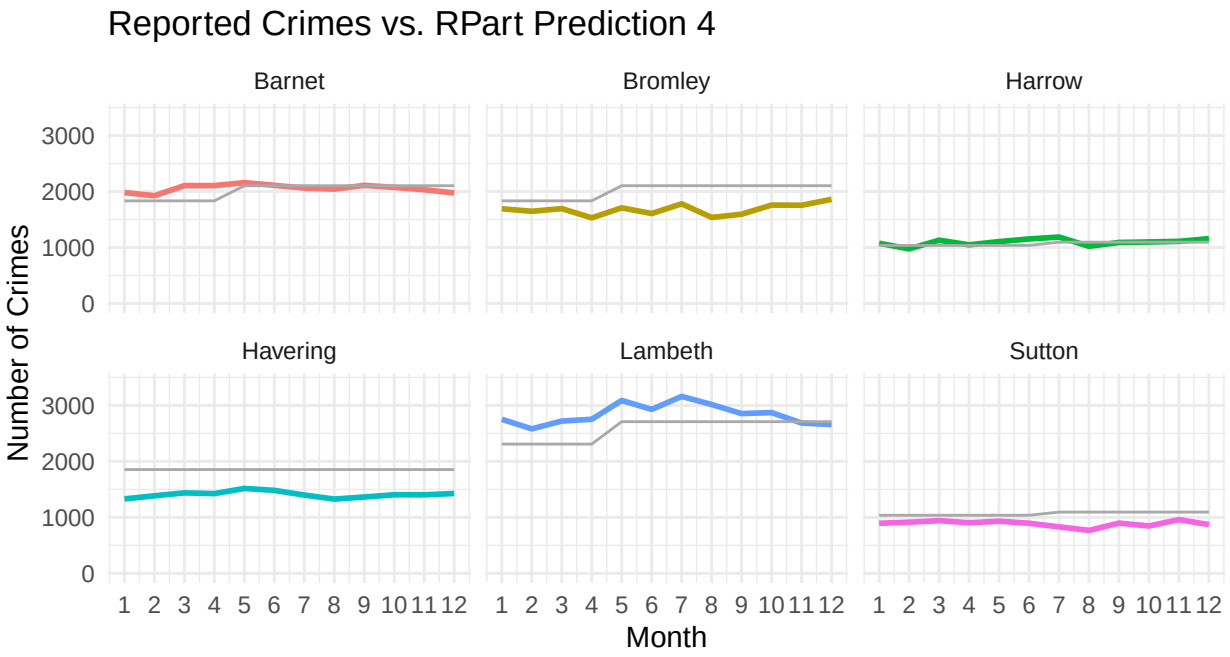
## Reported Crimes vs. RPart Prediction 2

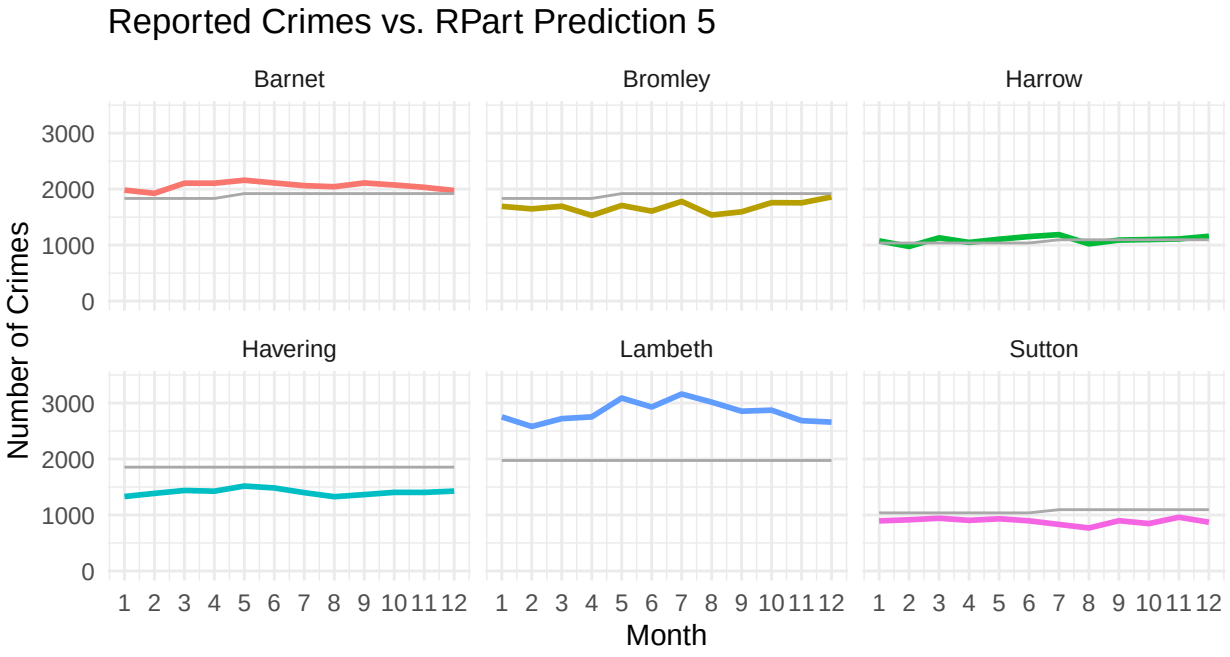



## Reported Crimes vs. RPart Prediction 3

(different y−axis than other charts!)



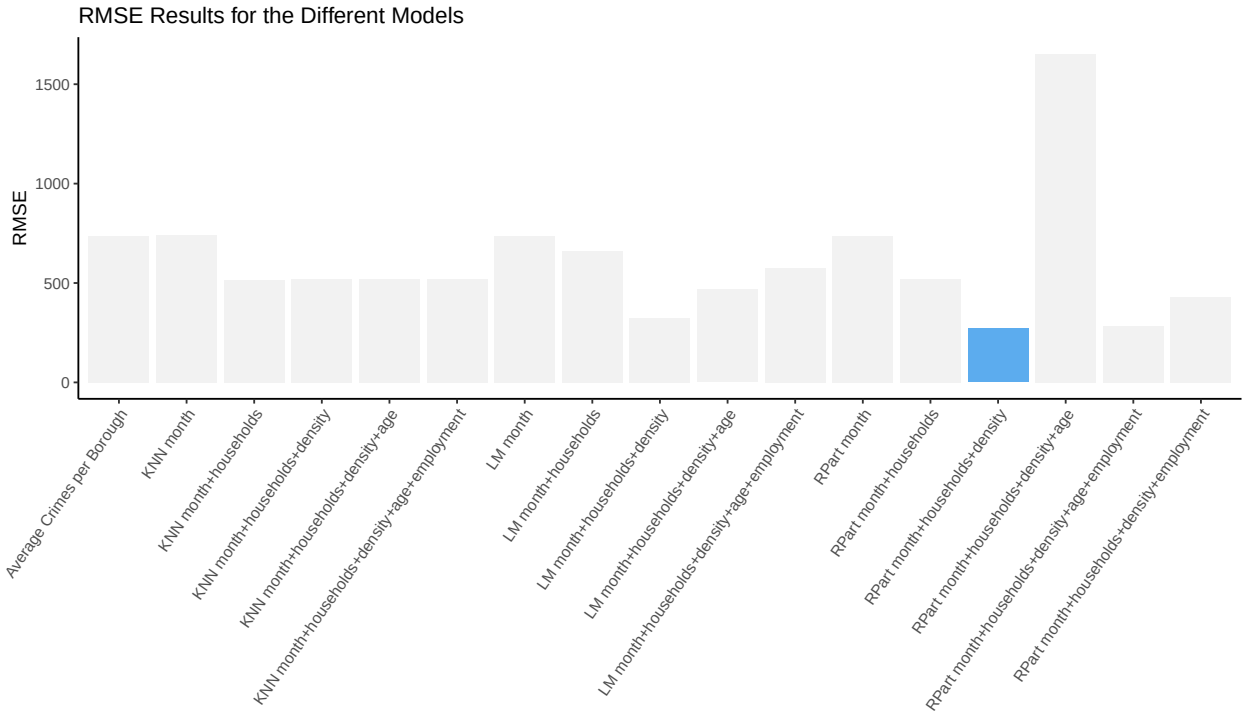



**Results of the Model Development**

In this chapter we have explored several different prediction models, starting with the simple Average and Linear Regression. Then we moved on to two more sophisticated methods with K-Nearest Neighbours and a Regression Tree. For all except the simple Average we have actually trained five different models. Several of the models performed well. The chart below illustrates the RMSE results of all models and highlights the best performing one: the Rpart Regression Tree with variables month + households + population density. In second place is the Rpart Regression Tree with variables month + households + population density + average age + employment rate male. The third best performing model was the Linear Regression with variables

month + households + population density.



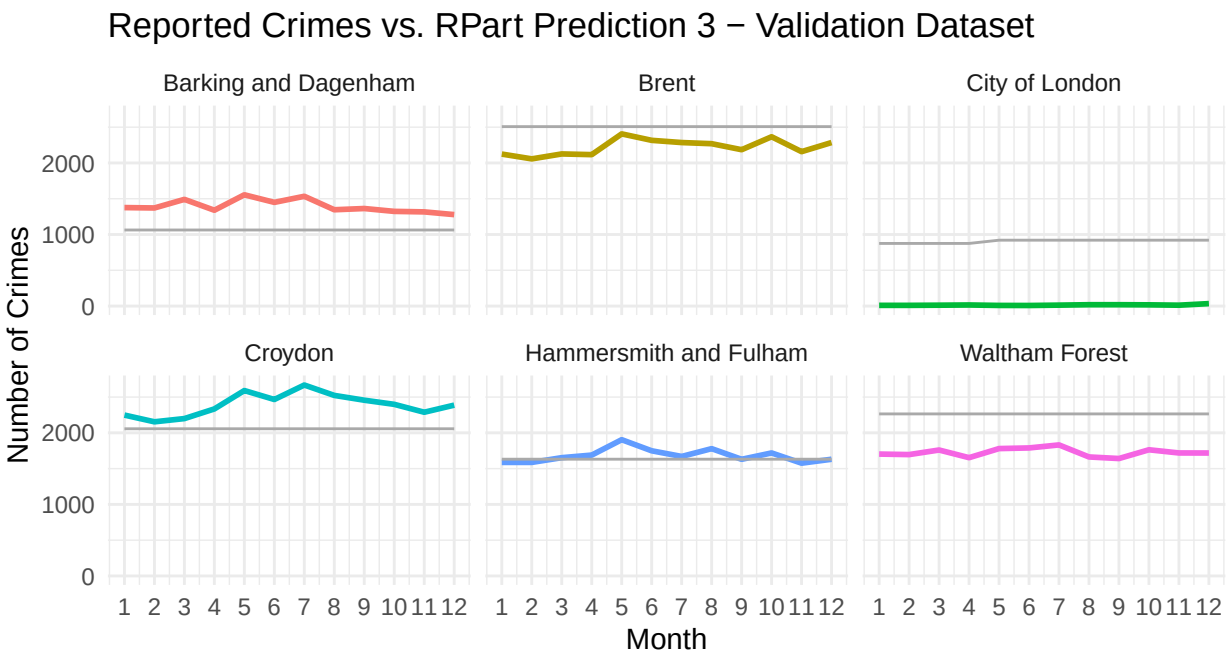
# 5 Results on the Validation Dataset

Now that we have identified our final model, we can evaluate the model performance on an unseen dataset, which is our validation dataset. In order to use as much training data as possible, we combine the original train and test sets to a new "enlarged" train dataset. This does not represent over-training, as we are not going to use the validation dataset for any model selection or tuning tasks. All we are achieving with this step is that we give our selected model with the selected best tune as much trainig data as possible, which hopefully improves performance on any unseen dataset. By combining the original train and test set, we give the model 27 boroughs to train on instead of the 21.

```
# fit the rpart model
# (using the rpart function instead of the train function, as we don't want to do any tuning,
# which the train function always includes, but use the best tune from rpart3)
fit_val_rpart3 <- rpart(crimes ~ month + household + population_density,
                        data = train_red_enlarged,
                        control = rpart.control(cp = fit_rpart3$bestTune))

# predict crimes in the validation set
predicted_crimes_val_rpart3 <- predict(fit_val_rpart3, newdata = validation_red)

# evaluate prediction for the validation set
rmse_val_rpart3 <- RMSE(reported_crimes_val, predicted_crimes_val_rpart3)
round(rmse_val_rpart3)
```

```
## [1] 490
```

With almost 500 the RMSE for the prediction of the validation set is definitely not as good as the prediction of the test set. However, this was to be expected, as we selected the best performing model for the test set. The chart below shows that the prediction is especially bad for the borough City of London. This borough is somewhat of an outlier as it has only about 10-30 crimes per month.




If we do the same prediction again but without City of London, the performance is much better (RMSE: 360). Naturally, we can't just drop one borough from our dataset because the prediction results are unsatisfactory. However, this result suggests that a further improvement of the prediction should be possible by forming a separate prediction for City of London, and potentially as well for Westminster. City of London has unusually low documented crimes and Westminster repeatedly turned up as an outlier during the exploratory data analysis. Maybe an approach with a combination of different forecasting methods could do more justice to those two boroughs. It would go beyond this project to add yet another prediction, but this could certainly be a starting point for future work.

# 6 Conclusion

Over the course of this project we have explored our dataset and throughly analyzed the different variables and their relationship with the number of crimes. The exploratory data analysis revealed that the number of crimes mostly varies depending on months, households, population density, average age and male employment rate. Using the insights from the EDA, we set out to predict the number of crimes for the boroughs in our test dataset. During model development we tried various different methods and model variations, ranging from the simple Average over Linear Regression to K-Nearest Neighbours and Regression Trees. The best performing model was the Regression Tree using the predictors month, households and population density which returned a final RMSE of 490 on our validation dataset.

However, predicting the crimes in our validation dataset revealed an issue: the borough City of London is an outlier for which the prediction is much worse than for the other boroughs. Addressing this issue should considerably improve the overall prediction results, as we have shown in the last chapter. Maybe an approach with a combination of different predictions for City of London and the other boroughs would do more justice to the differences. Another interesting extension - although not as promising as better accounting for the City of London - could be to try Random Forests. Random Forests address the shortcomings of Decision or Regression Trees, because they reduce instability by averaging multiple Trees. Basically, a Random Forest is "a forest of trees constructed with randomness" (source: PH125.8x Machine Learning). The Regression Tree was our best performing method, so further improving this approach could turn out to be beneficial.

# 7 Appendix

### Environment

```
## [1] "Operating System:"
```

```
##                _
## platform       x86_64-w64-mingw32
## arch           x86_64
## os             mingw32
## system         x86_64, mingw32
## status
## major          3
## minor          5.2
## year           2018
## month          12
## day            20
## svn rev        75870
## language       R
## version.string R version 3.5.2 (2018-12-20)
## nickname       Eggshell Igloo
```